Visualization and Analysis of Vortex-Turbine Intersections in Wind Farms

S. Shafii, H. Obermaier, R. Linn, E. Koo, M. Hlawitschka, C. Garth, *Member, IEEE*,
B. Hamann, *Member, IEEE* and K. Joy, *Member, IEEE*

Abstract—Characterizing the interplay between the vortices and forces acting on a wind turbine's blades in a qualitative and quantitative way holds the potential for significantly improving large wind turbine design. The paper introduces an integrated pipeline for highly effective wind and force field analysis and visualization. We extract vortices induced by a turbine's rotation in a wind field, and characterize vortices in conjunction with numerically simulated forces on the blade surfaces as these vortices strike another turbine's blades downstream. The scientifically relevant issue to be studied is the relationship between the extracted, approximate locations on the blades where vortices strike the blades and the forces that exist in those locations. This integrated approach is used to detect and analyze turbulent flow that causes local impact on the wind turbine blade structure. The results that we present are based on analyzing the wind and force field data sets generated by numerical simulations, and allow domain scientists to relate vortex-blade interactions with power output loss in turbines and turbine life-expectancy. Our methods have the potential to improve turbine design in order to save costs related to turbine operation and maintenance.

Index Terms—flow visualization, applications, wind energy, turbulence, vortices.

1 Introduction

Turbulence is a phenomenon that is believed to significantly affect a wind turbine's ability to harvest kinetic energy from the atmosphere. Additionally, it plays a critical role in the replenishment of kinetic energy in turbine wakes and is also thought to dictate the variability of the loads felt by the turbine blades. In the context of wind turbine arrays, where turbines are arranged in multiple rows, the performance and reliability of turbines behind the first row are strongly influenced by the combination of ambient turbulence and turbine-induced turbulence from upstream turbines. In order to maximize the benefits of wind energy and protect against the undesirable wear and tear caused by these ambient and turbine-related effects, it is critical to be able to identify, visualize, and study the various turbulent features that arise in this situation.

We present an integrated system developed for the analysis and visualization of data resulting from wind turbine simulations. Specifically, our research is motivated by the need to have powerful methods available to help study the interplay between wind fields and the blades of large wind turbines. The conditions to be analyzed include the turbine blades and the turbulence that could place spatially varying and transient loads on the blades' structures, possibly leading to decreased turbine life times.

One turbulent feature assumed to affect our turbines is known as a *vortex* or *eddy*, which intuitively resembles particles rotating around a common center [1]. There exist multiple definitions of a vortex based on various physical and/or geometrical properties [2]. To study how turbulence in simulated wind-turbine data sets affect turbines, we extract *vortex hulls* [3], where each hull describes a global feature that groups particles that exist inside of the same vortex.

These hulls are based on the reliable λ_2 [4] criterion and extract distinguishable global features. Furthermore, each vortex hull defines the region that contains each vortex core line and can be used to depict which portions of the turbine blades are intersected by vortices. We keep track of the sections of the blade geometry that are intersected, record simulated force values corresponding to these sections, and compute related force and intersection-based statistics. Since it is important to distinguish between vortices that are created by a turbine's rotation and "incoming" vortices created externally from ambient conditions or upstream turbines, we have developed a criterion that eliminates self-induced vortices.

This work makes the following contributions to the analysis of turbine-physics and visualization:

- Direct extraction and visualization of vortex-turbine intersections, where the vortex is not created by the turbine blade's rotation
- Statistical measurements of intersections versus blade geometry, and local force values
- A system for joint analysis of global wake behavior,

- *S. Shafii, H. Obermaier, B. Hamann and K. Joy are with the Institute for Data Analysis and Visualization, Department of Computer Science, University of California, Davis, CA 95616.*
- *R. Linn and E. Koo are with the Computation Earth Sciences Group (EES-16) at the Los Alamos National Laboratory, Los Alamos, NM 87544.*
- *M. Hlawitschka is with the Institute of Computer Science, Universität Leipzig, Germany.*
- *C. Garth is with the Computational Topology Group, University of Kaiserslautern, Germany.*

local vortex-blade intersections, and statistics

We introduce an integrated visualization tool that combines and semantically links the visualization of vortices, the depiction of vortex-blade intersections, and the analysis of said intersections so that domain experts can easily correlate structures of the flow field with observed effects on the blades. These three items are combined into one complete visualization and analysis system because it can be difficult and impractical to manually describe events that occur over long periods of time in simulated wind turbine data sets. Our visualization pipeline supports researchers with their task of identifying interesting events, and can help them draw conclusions about long-term behavior in a simulation.

The visualization problem is introduced in Section 2. The visualization system requirements and design of this problem are outlined in Section 3 and the corresponding implementation is described in Section 4. The results and evaluation are provided in Sections 5. Future research possibilities are covered in Section 6.

2 Problem and Motivation

The occurrence of early physical failure of wind turbine blades poses a considerable financial risk in the wind energy industry. Physical defects significantly increase a blade's operation and maintenance (O&M) cost, which is considerable compared to the cost of failure of other parts in a wind turbine system. It has been suspected that the localized extreme aerodynamic loads on the blade and some extreme mode of structural response, which cause the failure of the turbine blades, are related to turbulence induced by other wind turbines and atmospheric turbulence. However, it has not been investigated thoroughly because it is very difficult to integrate two physical processes – turbulence and aerodynamic load on the blade – with different length scales.

The data set used for this problem was simulated using HIGRAD/FIRETEC-WindBlade [5], obtained from Computational Earth Sciences Group (EES-16) at the Los Alamos National Laboratory (LANL). HIGRAD/FIRETEC-WindBlade is a coupled-physics atmospheric computational fluid dynamics model, combining Large Eddy Simulation (LES)-style turbulence ("HIGRAD/FIRETEC") with a Lagrangian representation of wind turbines ("WindBlade"). The HIGRAD/FIRETEC component has been under development at LANL since 1994 for a variety of applications including hurricanes [6], and is thus designed to account for the effects of complex terrain [7], surface roughness (vegetation) [8], and a wide range of turbulent length scales.

WindBlade [5] is a recent addition to HIGRAD/FIRETEC, and models wind turbine blades as Lagrangian particle elements to calculate aerodynamic responses of wind turbines, including aerodynamic load and turbine performances. HIGRAD/FIRETEC-WindBlade is designed to accurately simulate turbulent wind flow around wind turbines in order to investigate the effects of various complex environmental conditions, including the presence of upstream wind turbines on the wind turbine performances.

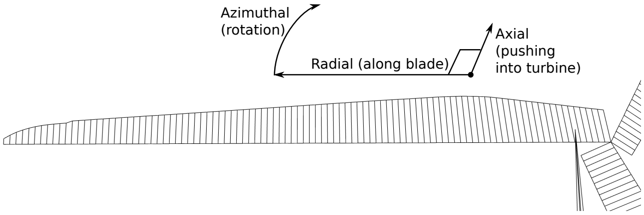


Fig. 1. A rendering of the turbine blade mesh with orthogonal force directions axial, azimuthal, and radial.

Therefore, it is a computational model designed to investigate the two-way interaction between operating wind turbines and the atmosphere by calculating forces on each blade in turbulent conditions. This simulation provides us with an incompressible, time-varying, three-dimensional flow field with embedded turbine blade geometry. These blades contain data related to physical quantities, such as forces acting on them.

The forces (or loadings) of the wind turbine blade result from a decomposition into three orthogonal directions: axial, azimuthal, and radial as shown in Figure 1. Radial force, which is parallel to the turbine blade pointing out from the hub, has almost no effect on wind turbine performance or structural response and is negligible. Azimuthal force is the force in the direction of the wind turbine blade rotation and induces rotation in the rotor system while axial force pushes the wind turbine rotor and causes most of the deformations and stress on the structure. The latter two forces are calculated per unit length along the blade, and represented by the unit N/m.

To study the turbine-turbulence interaction, domain scientists require a visualization solution that visualizes and characterizes the detailed interplay between the turbulence, represented as incoming vortices, and aerodynamic loadings on the blade. This integrated visualization system must depict the relationship between these vortices and the blades of turbines for simulated time-varying data, and must provide scientists a summarized analysis that was not previously available to them.

3 Requirements, Background and Design

3.1 Requirements

To study the relationship between blade forces and turbulence, a visualization solution must clearly depict individual vortices and the locations where the vortices strike the blade(s). Vortices are prominent features of turbulence in wind flow fields, and it is expected that the interplay (or intersections) between these vortices and the turbine blades provides insight into the potentially damaging effects on the blades. Each vortex must be numerically stable, have a quantifiable region of influence, and must be created externally (i.e. each is not induced by the blade's rotation, and is referred to as an "incoming" vortex). Since it is expected that many vortices exist in the region close to the blade geometry, it is challenging to display the blades and vortices while providing information regarding their interaction.

Additionally, information is required regarding how frequently certain blades are struck by vortices during the simulation. Since the turbine blades contain simulated scalar forces, the force values residing in those intersected regions must be recorded. Numerical analysis of these intersections and their force values allows one to establish a relationship between the vortices and blade geometry.

Furthermore, this qualitative and quantitative analysis must be conducted on individual blades of each turbine. Since there exist many time steps in a simulation, statistics gathered during the visualization process should be visible for each time step and aggregated over all time steps so that we are able to deduce peak as well as integrated loads. Tracking vortices over time is not required, as there is no interest in which vortices a blade is struck by; rather, all intersection points as apparent over all time steps are of interest. In summary, a suitable visualization system should satisfy the following requirements

(1) Visualization of global turbine and (incoming) vortex behavior:
   a) Visualization of turbine geometry
   b) Visualization of individual hull-like vortex features
   c) Visualization of turbine-vortex intersections in real-space
(2) Visualization of local turbine-vortex interaction:
   a) A clear depiction of vortex-blade intersections and identification of interesting events
   b) Relationship between intersections and the following: force value, geometry
   c) Depiction of single and multiple time step behavior to deduce a relationship between the vortices and turbines
(3) Visualization of statistics regarding intersections in relation to blade geometry and force values on the blades

3.2 Background on Vortex Extraction Techniques

Since we require an extraction technique to visualize the turbulence that affects our simulated wind farms, it is necessary to give a brief summary of established techniques. Most of these extraction methods can be categorized as *line-based* or *region-based*. While the former type of method searches for line-like features representing vortices, the latter type identifies contiguous data set regions that belong to a vortex or vortex core [2]. Common region-based extraction techniques employ local operators, and include the λ_2 [4], $\mathbf{Q}$ [9], and $\boldsymbol{\Delta}$ [10] criteria.

We require line-based vortex methods that identify global features and a region of coverage (or hull) associated with each feature, as the vortex hulls could be very close to the blades and potentially intersect them. Among line-based techniques, the method described by Roth and Peikert [11] detects vortex core lines by identifying areas in a data set where the derivative of the acceleration with respect to time is parallel to the velocity vector. The approach described by Kenwright and Haimes [12] detects vortex cores by solving for the eigenvalues and eigenvectors of the velocity-gradient tensor ($\nabla \boldsymbol{u}$), where the eigenvectors corresponding to the pair of complex-conjugate eigenvalues describe the plane containing spiraling flow and the eigenvector corresponding to the real eigenvalue points in the direction about which the flow spirals.

The technique employed by Sujudi and Haimes [13] finds the cores of "swirling flow" by searching for points where $\nabla \boldsymbol{u}$ has one real and one pair of complex-conjugate eigenvalues, and where the "reduced velocity" (calculated from the velocity and the eigenvector corresponding to the real eigenvalue) is zero. They compared their method against one found in the "FAST" [14] visualization environment, where vortex cores start from spiraling saddles. Garth *et al.* [15] presented a method that extracts vortex core lines from assumed vortex core positions and renders a surface around the core to verify its existence.

There also have been attempts to detect vortices as ridge or valley lines of scalar indicator fields such as λ_2, Q-criterion [16], and others. Sahner *et al.* [17] accomplish this task by using higher-order derivatives, while a similar method discussed in a paper by Sahner *et al.* [18] extracts extremal regions (skeletons and sheets) relating to vortical and high-strain regions using the first derivative. An alternative approach to finding extremal lines of vortices following (minimal) pressure scalars and vorticity vectors, named the "predictor-corrector" technique, was developed by [19], [20], [21]. It was extended by [3], [22], [23], [24], where the λ_2 criterion is used instead of pressure, and vorticity vectors are retained for vortex directions.

For a survey of various vortex detection methods, one can refer to the work by Jiang *et al.* [2] and Post *et al.* [25].

3.3 Design

Our visualization and analysis pipeline to realize the requirements stated above consists of four stages as shown in Figure 2.

Initially, we render vortex hulls from our time-varying data set, and identify the subset corresponding to incoming vortices. Vortex hulls are global features identifying individual vortices, and are preferable to standard isosurfaces of a λ_2 threshold, which represent vortices as well. A comparison between hulls and isosurfaces of λ_2 is shown in Figure 3. Each hull defines the region containing a vortex core line and allows us to test for intersections against blade geometry in stage (b) of Figure 2. The intersection regions give a direct clue of turbine-vortex interaction while reducing visual cluttering when rendering the cores and blades in the same view. These portions of the pipeline satisfy the first requirement stated above.

Since it is difficult to visually keep track of the blades and their intersections because the blades are continuously

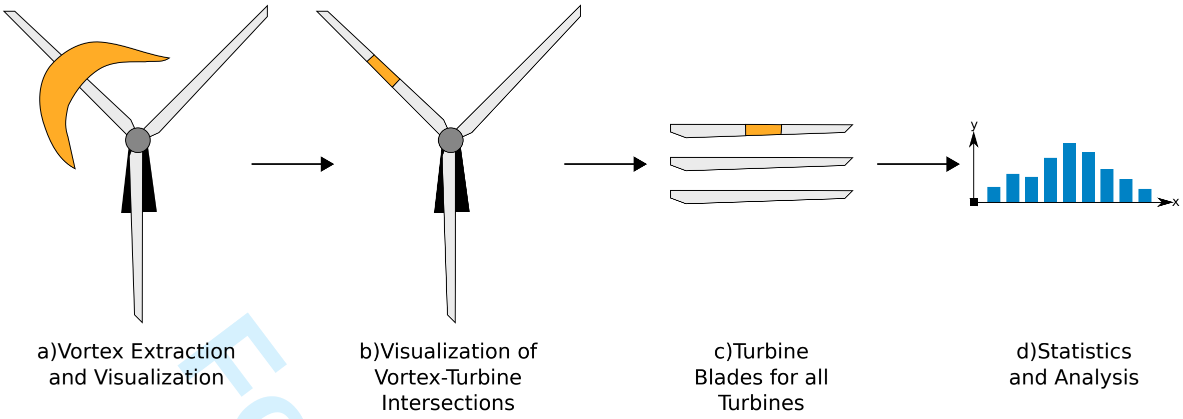
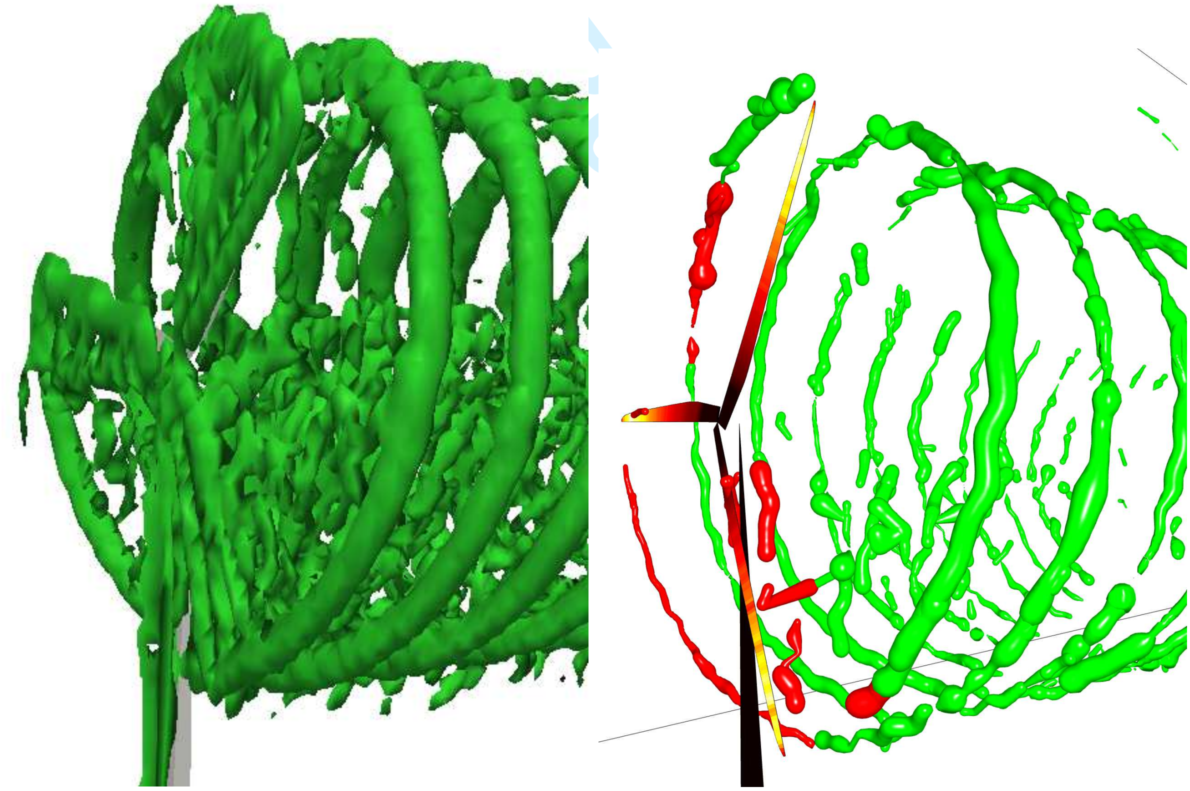

a)Vortex Extraction and Visualization
b)Visualization of Vortex-Turbine Intersections
c)Turbine Blades for all Turbines
d)Statistics and Analysis
y
x

Fig. 2. The four main stages of our visualization pipeline, illustrated by four items of this flow chart (stage "a" through "d"). After the data is read in, the vortices are extracted and visualized as features that exist inside a simulated wind turbine farm (a). The artificial vortex in this figure is illustrated as a horseshoe-shaped orange hull. In the next stage (b), the intersections of the vortex hull(s) and the blade(s) are extracted and visualized. To allow a scientist to compare blades in a consistent fashion, the blades are oriented horizontally in a static blades view (c). These intersections are then used to gather and analyze statistics (d) to establish a relationship between the vortices and blades. A graph view is created for the statistics available in each time step.



(a) Vortices extracted by standard λ_2-based isosurfaces.

(b) Vortex hulls created using the technique described in [3]. Vortices in vicinity of turbine are colored red.

Fig. 3. A comparison of vortices extracted as isosurfaces of a λ_2-threshold in 3(a) and vortex hulls in 3(b). It is easy for us to control the amount of features extracted while generating vortex hulls, whereas it is difficult to do the same thing with plain isosurfaces of λ_2 when most of the important vortex features are rendered. Furthermore, we can isolate individual features among the vortex hulls, and color them based on proximity to vortex turbines as indicated by the red vortex hulls in 3(b).

rotating from time step to time step, we keep their positions fixed in a static blades view, as shown in stage (c). This type of visualization solves the issue of visual inconsistency between time steps, as consistent blade identification in a rotating blade turbine is challenging or even impossible for high angular velocities. In case there are multiple (upstream and downstream) turbines in a data set, a domain scientist can conveniently visualize and compare the blades of all turbines and their intersections in this view. We identify the sections of the blade geometry that are intersected, and the force values that exist in those sections, satisfying the intersection portion of the second requirement for our visualization system (2)(a). From this information, we compute the following statistics (stage (d) of pipeline):

(1) Number of intersections as a function of force value for each turbine blade
(2) Number of intersections as a function of blade radius for each turbine blade

These statistics permit direct numerical examination of the blade intersections, bridging the gap between symbolic and visual data analysis. To that end, statistics can be viewed in an auxiliary graph view while a time step is being visualized. In addition, the program can apply our visualization pipeline and compile data from time step to time step automatically, allowing one to aggregate and study the resulting statistics afterwards. The computed statistics satisfy the third requirement and (2)(b) while the aggregate statistics identify multiple time step behavior as specified by requirement (2)(c).

The composition and semantical linking of the above visualization components allows for a concurrent analysis of joint vortex and blade behavior, all of which is visible inside of one system.

4 System Implementation

These sections describe the implementation of our visualization system as introduced in Section 3. We discuss the implementation of vortex core extraction and visualization (section 4.1), intersection testing of vortex hulls with blade geometry (section 4.2), static turbine blades view (section 4.3), and computation of relevant statistics (section 4.4). Screenshots of the different windows for our visualization system is shown in Figure 4.

4.1 Vortex Extraction and Visualization

In an effort to extract coherent vortical features, we make use of a *predictor-corrector* method [3], as it is capable of extracting the vortex hulls that we require and is based on the robust λ_2-criterion without sharing disadvantages of classic isosurface techniques. First, seed points are created by identifying grid points that have negative λ_2 values and are local minima (since the more negative λ_2 is at a point, the stronger the vortex). From these seeds points, one integrates vortex core lines based on the interpolated vorticity vector, and corrects each integration point to minimize the value of λ_2 in the plane perpendicular to the integration direction. We apply correction using a derivative-free "direct search" approach [26] to accurately recreate a vortex core line during integration. Since convergence is not guaranteed, the vorticity direction at this point must not differ excessively from the predicted version.

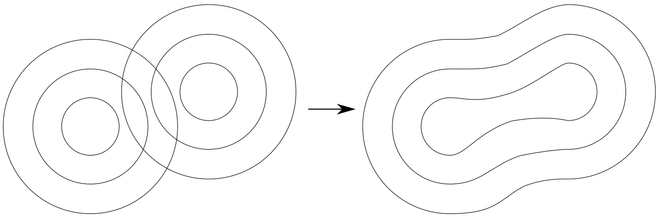


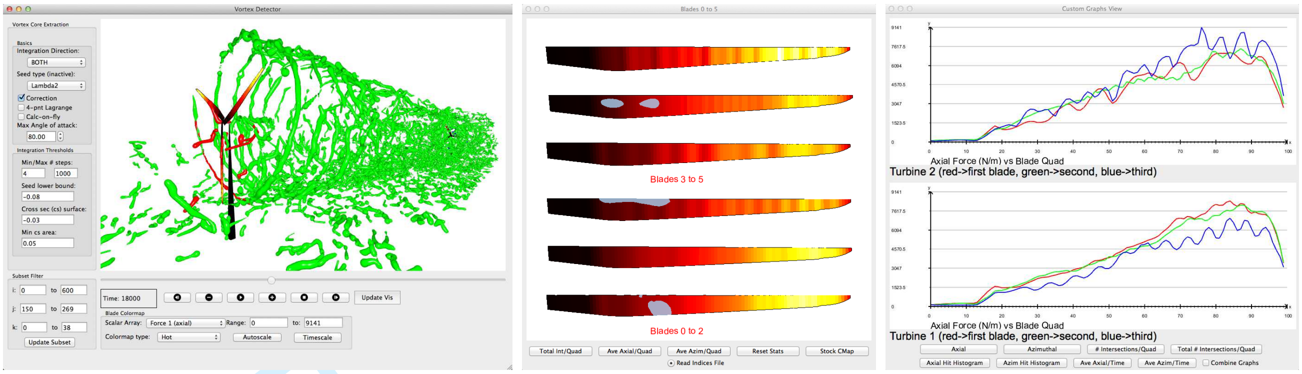
Fig. 5. Sketch of a two-dimensional metaball. On the left, two isotropic kernels are shown by a selection of their respective isocontours. On the right, we add the kernel values of the two isocontour sets, and create contiguous isocontours. In 3D, this yields a smooth metaball-based vortex core hull if the kernels are placed along vortex core lines.

During integration of the vortex core lines, the predictor-corrector algorithm keeps track of the hull's cross sections by casting sample rays in the plane spanned by the vortex direction. Every sample ray stops once a λ_2-based threshold is violated, or if there is a large angular difference between the vortex direction and the interpolated vortex direction at the current ray's sample position. The size of each ray is used as a radius value and the radius values from all sample rays can be used to calculate area. Integration continues as long as the current cross-sectional area is large enough, as long as a predefined integration limit is not violated, and as long as the vortex core line stays within the bounds of the data set. Since the radius values of the sample rays can be periodic, Stegmaier et al. [3] used a discrete Fourier series to store and re-create the cross sections as discrete segments of polygonal vortex hulls.

Because in our application these cross sections are mostly isotropic and standard creation of polygonal tubes may often lead to irregular polygons and self-intersections in the case of rapidly changing core directions, we extract our hulls using metaball surfaces [27], [28]. In order to generate water-tight, self-intersection-free vortex hulls, we superimpose an additional scalar field onto the vortex core line visualization. First, individual core lines are sampled by position and cut-off radius sample pairs (x_i, r_i), with step sizes corresponding to their local hull (average) radius r_i. At every such sample, we add a rasterized isotropic kernel

$$w(x,x_i) = \left(1.0 - \frac{\|x-x_i\|^2}{r_i^2}\right)^3 \qquad (1)$$

to the superimposed scalar field, if the value of $w(x,x_i)$ is positive. We have chosen this specific function $w(x,x_i)$ as a kernel as its behavior is quite similar to that of a Gaussian function and one can compute the $w(x,x_i)$ values more



Integration Direction: BOTH

Seed type (inactive): Lambda2

☑ Correction

□ 4-pnt Lagrange

□ Calc-on-fly

Max Angle of attack: 80.00

Min/Max # steps: 4 1000

Seed lower bound: -0.08

Cross sec (cs) surface: -0.03

Min cs area: 0.05

i: 0 to 600

j: 150 to 269

k: 0 to 38

Time: 18000

Scalar Array: Force 1 (axial) Range: 0 to: 9141

Colormap type: Hot

● Read Indices File

□ Combine Graphs

Axial Force (N/m) vs Blade Quad

Turbine 2 (red->first blade, green->second, blue->third)

Axial Force (N/m) vs Blade Quad

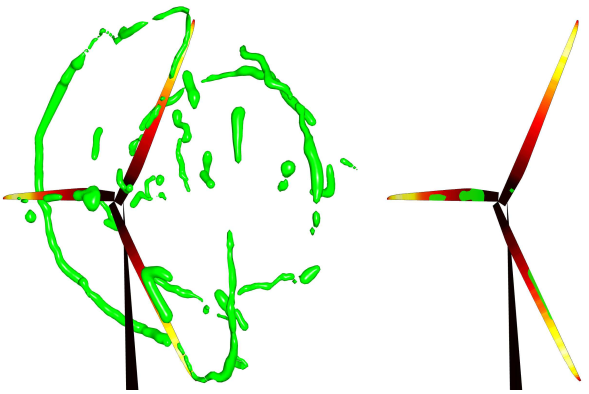
Turbine 1 (red->first blade, green->second, blue->third)

(a) Main window showing visualization of wind turbine and vortices.

(b) Window with static visualization of turbine blades.

(c) Window showing graphs of statistics.

Fig. 4. The three windows of our integrated visualization system. Figure 4(a) is a rendering of the wind turbine and vortices in the same space. The extraction of vortices is explained in section 4.1. In Figure 4(b), we show the window that displays the static blades view and the intersection regions highlighted in a blue. The buttons control which colormap is rendered onto the blade surface (see section 4.3 for an explanation). Finally, the graphs view in 4(c) displays statistics for the data set, and is described in section 4.4.



(a) Extracted vortices and turbine.

(b) Intersections of vortices and turbine geometry.

Fig. 6. Visualization of (green) vortex hulls and turbine in 6(a) and relevant intersections (colored in green) in 6(b). The vortex hulls are rendered using metaballs, which recreate the vortices as smooth features that do not suffer from self-intersecting geometry.

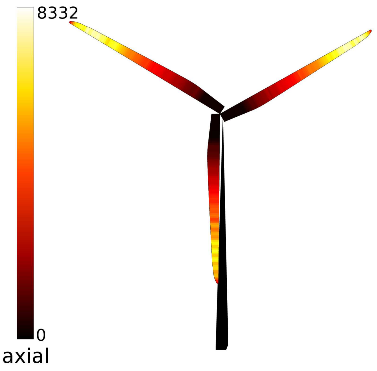


Fig. 7. A rendering of the turbine blade mesh using a colormap (used throughout this paper) of axial force. Wind turbines in our data sets consists of three blades of no thickness, where each blade is composed of one hundred quadrilaterals as portrayed in Figure 1, and a tower (shown standing behind the blade triplet). Each quadrilateral contains a simulated axial and azimuthal force value. Higher force values are indicated by brighter colors that are close to yellow, while weaker force values reside in regions of darker color.

efficiently than values of an exponential Gaussian function. The kernel's infinite extent is limited to the cut-off radius to ensure fast generation of the scalar field. A two-dimensional example is shown in Figure 5. This rasterization process creates a scalar field representation of the vortex hulls that closely resembles a smoothed distance transform. After adding this scalar field, we extract isosurfaces of this scalar field to create closed, smooth vortex core hulls, whose radii correspond to the locally computed vortex hull average radii. A sample visualization is shown in Figure 6(a).

4.2 Visualization of Vortex-Turbine Intersections

After extracting vortex cores, we visualize their interaction with the turbine blades. In the data that we have used in our examples, there exist turbines with three blades each. Therefore, when there are two turbines we use blade indices from zero to five. Each blade's geometry is represented as a series of one hundred quadrilaterals (or cells) along its main or radial axis as seen in Figure 1. A blade quadrilateral has zero thickness and contains information regarding simulated forces that exist on it. An illustration of turbine blade geometry is shown in Figure 7.

We calculate intersections between the vortex hulls and turbine blades by using shadow volumes [29]. In our case, the shadow volume corresponds to the set of extracted vortex hulls, and the turbine blades are the objects that receive the "shadow" created by these hulls. By using the OpenGL stencil buffer to define regions inside of the volume, we isolate portions of the blade geometry that are intersected by the hulls. An example is shown in Figure 6(b).

We are only interested in external vortices that may affect a turbine's performance, or vortices that are not self-induced by a turbine blade's rotation. Based on our observations, there exist *bound vortices*, that are attributed to the lift force of the blades [30], and *tip vortices*, which are created at the tips of the blades because of a difference in pressure between the lower and upper side of each individual blade [30], [31]. These self-induced vortices are adjacent to the blades from which they originated, and their axes are parallel to each turbine's plane of rotation. A visual example is shown in Figure 6(a), where the bound vortices run down the length of the blade and tip vortices are created at both of the blades' tips. Over time, tip vortices gain a translational component that is perpendicular to the plane of rotation and travel downstream. Since these self-induced vortices are present in both turbines and could negatively influence the statistics, we only calculate intersections of hulls that are not parallel to this plane of rotation – or hulls that are created externally. To test for this condition, we compute the average direction of the portion of each hull that exists inside of a small bounding box surrounding each turbine. If the angle between the average direction and plane of rotation indicates that the vortex is not generated by the turbine, the vortex hull is identified as one that strikes the blade.

An angle value of less than ten degrees was sufficient to eliminate self-induced vortices. The resulting intersection regions of the striking vortices are stored in an offscreen buffer for later use.

4.3 Static Turbine Blades View

After finding the intersections between the blades and vortex hulls, it is difficult to perceive the time-varying intersections in a consistent manner as the blades are continuously rotating during the simulation. To make it easier to view the blades in a constant position that allows time-varying analysis of intersection locations and force values, we create an auxiliary blades view in which each blade remains in a horizontal orientation as shown in Figure 8. Since our simulations may have multiple turbines, each corresponding to a set of blades, the view is created for all blades with annotations indicating each set of blades. The (force-based) colormap for these blades is adjusted to match the counterpart in the main ("real-space") visualization. Each vortex is individually intersected against the blade geometry using the stencil buffer in this view using the method described in Section 4.2.

Unlike the main visualization, the static blades view provides a consistent rendering of the blades over time as their order and position stays fixed. This allows a scientist to easily compare the intersections of each blade for every time step of the simulation and draw conclusions about the relationship between the vortices and the blades' force values. For example, if the downstream turbine is intersected by more vortices than the upstream turbine, the static blades view will clearly depict this event and show which values of force correspond to the intersections for both turbines.

Furthermore, this view portrays average axial force, average azimuthal force, and total number of intersections per blade quadrilateral accumulated for all time steps processed so far. These scalar quantities can be visualized on the blade using a colormap, allowing one to see which portions of the blades have received the most intersections up to a certain point, and which parts of the blade tend to have the strongest force values.

When used in conjunction with the real-space visualization, the static blades view is useful for analyzing temporally large data sets, where it is difficult for a scientist to examine each time step's blade geometry using a standard visualization of vortices and turbines.

4.4 Statistics Computation

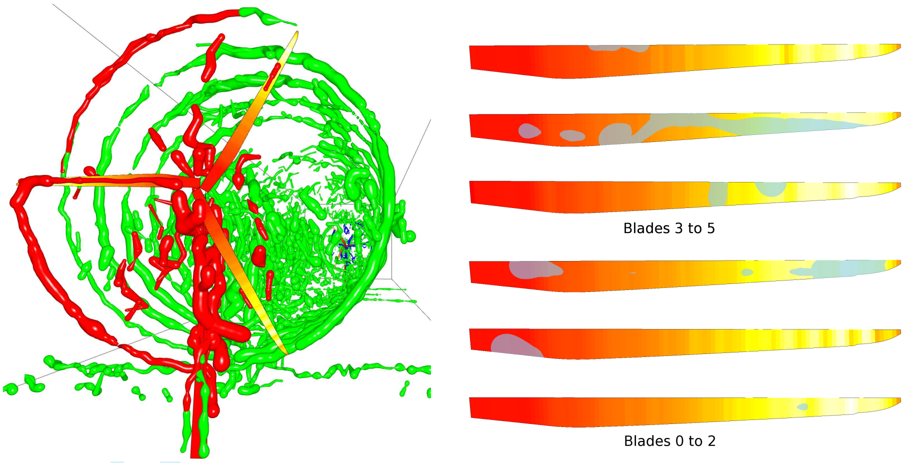
After extracting our intersections visually, it is necessary to identify which sections of the blade correspond to the intersections, and the magnitude of local force values. The stencil buffer alone indicates which pixels are covered by the vortex hulls, but it does not indicate which blade quadrilaterals are intersected.

To solve this problem, we use our static blades view, which keeps the blades in a constant resolution and orientation. We perform an initial (off-screen) rendering pass which colors the intersected quadrilaterals by converting their one-dimensional compound quadrilateral index to a (R,G,B) color triple. This compound index identifies the global blade index and the quadrilateral index local to that blade. Therefore, when there are two turbines with three blades each, the blades are numbered with identifier $b_i \in [0,5]$. Thus, the compound index c_i is represented as $c_i = b_i \times 100 + q_i$, where q_i $(0 \leq q_i \leq 99)$ is the index of the quadrilateral local to the blade.

After each vortex hull is intersected against the blades using the stencil buffer, only parts of the blades that are intersected by the hull are rendered. Invalid intersection pixels have a color of (0, 0, 0) (blade border) and (255, 255, 255) (background) or black and white, respectively. This offscreen pixel buffer is processed in parallel on a GPU, marking the blade quadrilaterals as intersected by converting the three-dimensional pixel buffer color value back into its one-dimensional (quadrilateral) compound index counterpart. There is a limited amount of skew in the blade geometry and the viewport is large enough to represent all of the blade quadrilaterals.

From this set of intersected quadrilaterals we keep track of the number of intersections per blade quadrilateral, and look up each quadrilateral's force value. This step bridges the gap between the visual representation of the intersections and the numerical analysis of force values.

We use the quadrilateral set to create a graph visualization view, which renders these intersection statistics for the



(a) Real-space vortices and turbines. It is difficult to see the blades of the rear turbine clearly.

(b) Custom blades view with intersections rendered as bluish regions. Blades 0 through 2 belong to the first turbine; 3 through 5 belong to the second.

Fig. 8. An example of how the custom blades view shown in 8(b) help us easily visualize the vortex-turbine intersections as visualized in 8(a). The custom blades view's colormap corresponds to the type used in Figure 7. In 8(a), most vortices are rendered using green unless they are inside a thin bounding box centered around the front turbine (red vortices) or rear turbine (blue vortices). Unlike the main visualization, the custom blades view permits a convenient visualization of the vortex intersections from a single perspective.

front and rear turbines as a line plot of intersection count versus the blade quadrilateral local index q_i, allowing the user to view real-time statistics as portrayed in Figure 9(a). If the user clicks on the graph, the graph view displays the corresponding blade radius value and highlights that portion in the blades view. Furthermore, it plots the average force values over time and allows the user to select a time step to investigate interesting behaviors in all views, such as spikes in force values. It is possible to overlay the graphs corresponding to the front and rear turbines, allowing one to visually compare the two turbines' statistics.

Our implementation externally stores statistics corresponding to intersection count as a function of blade quadrilateral or force value, which allows us to not only discover interesting events while visualizing individual time steps but also aggregate results during post-processing.

5 Results

In the following we analyze the applicability of our visualization system to a two-turbine data set and evaluate its usefulness with respect to the scientific insights gained.

5.1 Data Set

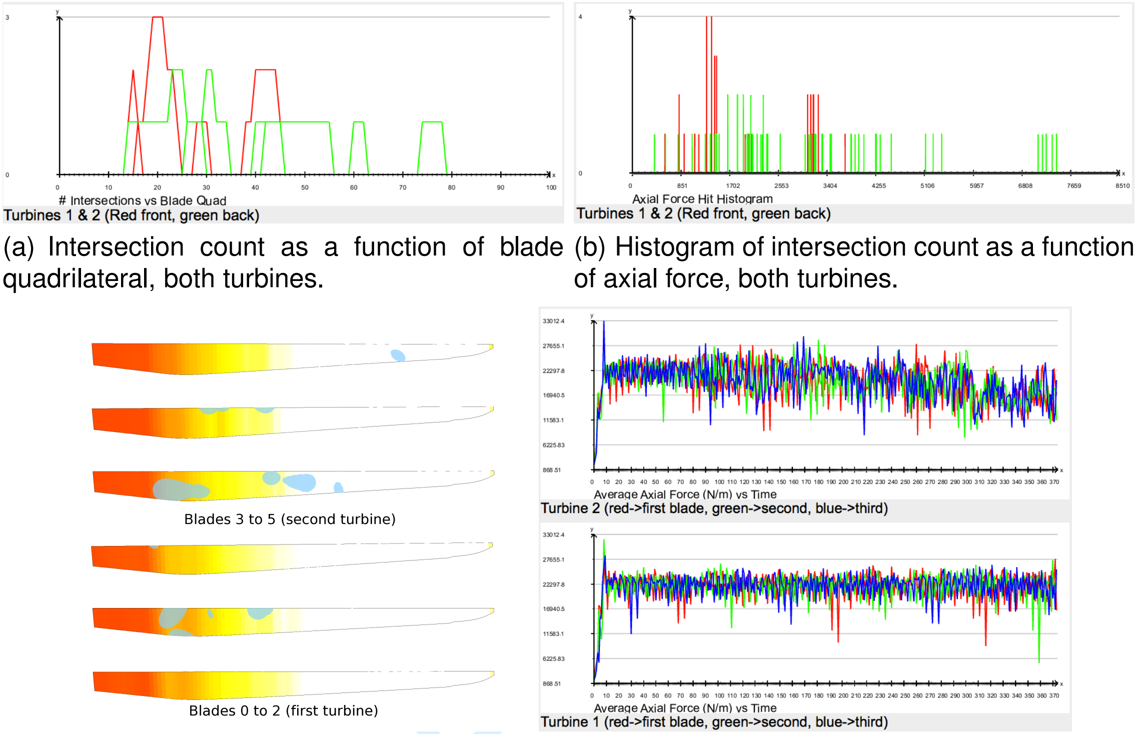
The data provided to us consists of 374 time steps (374 seconds), and is 2160 meters long, 840 meters wide, and 300 meters tall. Since we are interested in turbulence conditions closer to the turbines, we clip away regions that are far away. The upstream and downstream turbines are occasionally referred to as the first and second turbines, respectively. Each turbine's blade is 63 meters long, which means that each blade quadrilateral is 0.63 meters wide.

The data set is initially non-turbulent however a low-level turbulent jet stream, also referred to as a high-speed nocturnal jet, propagates toward the top of the turbines from the inlet boundary. This input stream models low-level jet conditions in the Great Plains region in the United States, and it exists in the altitude range of 250-450 meters, with a peak of 350 meters. In general, these conditions correspond to high atmospheric turbulence induced by high wind shear (the vertical gradient of wind speed), and often cause the down time of wind turbines. For this simulation, the time-average hub-height (90 meter) wind speed is 14.62 $meters/second$, and turbulent wind conditions is generated based on turbulent data measured in Lamar, Colorado [32]. While the low-level jet does not directly strike and interact with the turbines, it does affect the turbulent flow that exists in the atmosphere.

After the flow field has adjusted to its initial conditions, the turbines begin to spin up due to the existing turbulent flow. Sample screenshots of this data set rendered using our visualization pipeline shown in Figure 10. It is clear from these screenshots that the second turbine experiences more turbulence compared to the first turbine, and the atmosphere close to the turbine contains more vortices as well. This phenomenon is attributed to the turbulence wake which is induced by the first turbine's rotation, and which propagates downstream to the second turbine. We analyze the turbulence that is created by the upstream turbine's rotation and observe how it affects its counterpart.

5.2 Results and Evaluation

Prior to analyzing the data set with our visualization system, the domain experts expected that there existed some



(a) Intersection count as a function of blade quadrilateral, both turbines.

(b) Histogram of intersection count as a function of axial force, both turbines.

(c) Visualization of the intersections in 9(a) on static turbine blades.

(d) Average axial force as a function of time.

Fig. 9. The graph visualization view for the front (“Turbine 1”) and rear (“Turbine 2”) turbines allows the user to view real-time statistics while navigating a time-varying data set. We render intersection count versus blade quadrilateral in 9(a), intersection count versus axial force in 9(b), and the corresponding static blade visualization in 9(c) for a specific time step. The blades view is a concise representation of the intersection and force statistics, allowing time-varying analysis of vortex-blade interactions. The second turbine has more intersections in areas of high force and high radius values relative to the first turbine. To visualize time-based behavior, we render average axial force versus time, which indicates a large oscillation of values in Turbine 2 near time step 90 in 9(d). The user can click an x-value on the time graph to navigate to a specific time step in case unusual behavior is noticed.

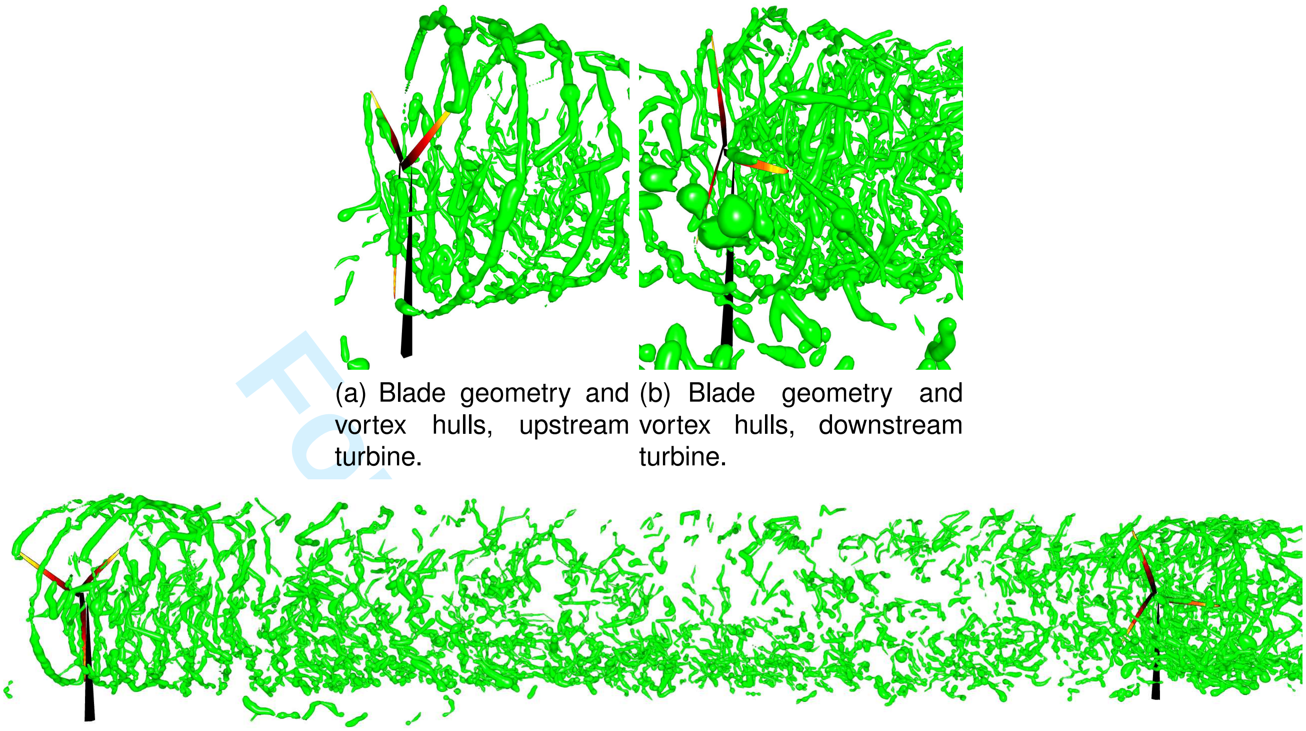
relationship between the turbulence in the data set and the performance of the downstream turbine. The application scientists on our team had observed a drop in the power output (measured in Watts) of the second turbine close to time step ninety in the simulation, as illustrated in Figure 11.

With our visualization system, it became clear that this change in power output corresponds to the turbulence that is created by first turbine and that traveled downstream toward the second turbine. Our visualization system depicted this result, as seen in Figure 12(f), and the unusual change in axial and azimuthal force values shown in Figures 12(b) and 12(d), respectively. The force values for the second turbine appear to oscillate during later time steps, and these oscillations are related to fluctuations in blade loading. Fluctuations in blade loading consequently lead to fatigue in the blade structure, an increase in maintenance cost and a decrease in turbine life-time. The real-space visualization of our pipeline confirms previous findings related to power output and maintenance and relates them to coherent features of turbulence and statistics.

Furthermore, there appears to exist a relationship between the power output and the occurrence of intersections at certain portions of the blade geometry. This behavior is exhibited in Figure 13, which displays intersections as a function of blade quadrilateral accumulated for all time steps as a graph, and displays a related static blades view that uses a color map based on the same statistic. The second turbine’s blades receive a greater number of intersections in the portion of the blade geometry that corresponds to higher radius values. The turbine’s power output is related with torque, which is defined by the expression:

$$\tau = r \times F \tag{2}$$

where r is distance (radius value) from the turbine’s rotor and F is azimuthal force. This means that the vortex intersections correspond to areas of the blade that are responsible for generating large amounts of torque. A blade that does not experience turbulence in these regions would not ordinarily suffer from power output loss. However, the rear turbine does show more intersections in the second-half of each blade (relative to the first turbine), and it does suffer from power output loss. This is a new scientific insight into



(a) Blade geometry and vortex hulls, upstream turbine.

(b) Blade geometry and vortex hulls, downstream turbine.

(c) Side-view rendering of wind farm simulation.

Fig. 10. Upstream turbine, downstream turbine, and side-view renderings of our test data set as shown in 10(a), 10(b), and 10(c), respectively. Although the vortices dissipate when traveling downstream, our visualization system permits us to analyze global wake behavior and observe events including the vortex wake reaching the second turbine. Furthermore, the second turbine's vortices appear to be more turbulent compared to the vortices generated from the first turbine. This is a direct effect of the turbulence induced by the first turbine's rotation.

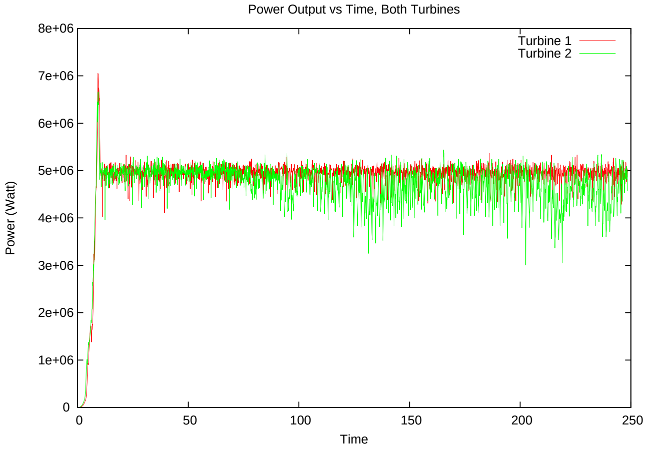


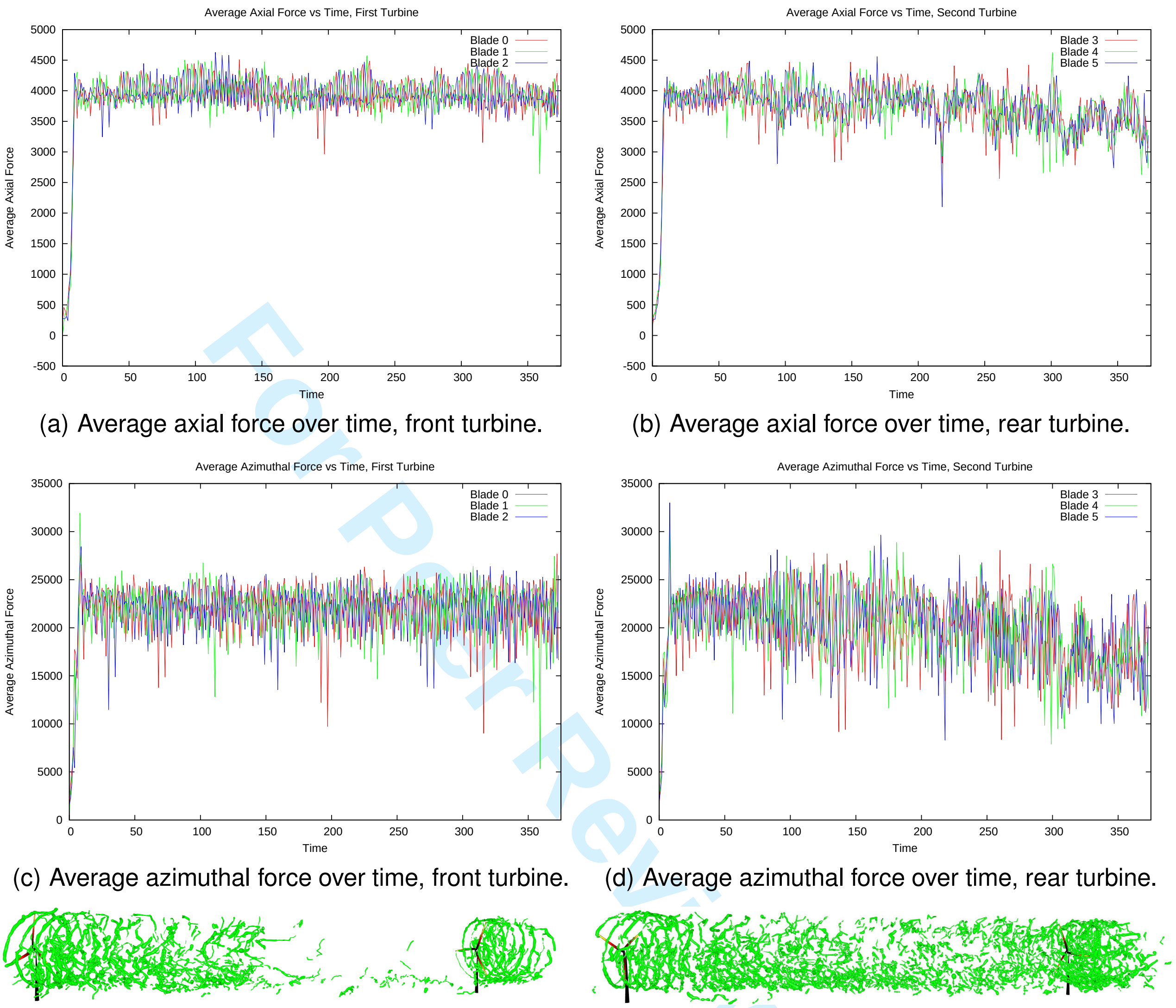
Fig. 11. Graph of power output for first and second turbines represented by the red and green lines, respectively, as obtained from our domain scientists. The number of time steps shown here is smaller than the amount in the results section, but it is sufficient to show a drop in the second turbine's power around time step 90. This occurrence correlates with the vortices striking the second turbine and the sudden change in its force values as shown in Figure 12.

the physical causes of the loss of power output that was not immediately well-understood by our application scientists, and is now visible to them in the statistics and the static blade views. Thus, semantic linking of the statistics, blade view, and real-space wake view makes possible effective and important visual correlation between power output loss and its causes.

We also investigated the relationship between the intersections and force value, and graphed a histogram of intersections as a function of azimuthal and axial force in Figure 14. We observe that the first and second turbines tend to have similar numbers of intersections in relation to force value. The second turbine receives more intersections in regions of high axial force, which hints towards a relationship between the intersections and high force value for the second turbine. Since axial force is related to stress on the turbine structure, these intersections could be related to damage on the blades. We plan to analyze a more complex and detailed simulation in the future that measures the turbines' physical stress so that this statistic is more thoroughly analyzed.

6 Conclusions and Future Work

Our visualization system represents a significant step toward a complete analysis of a simulated wind turbine



(a) Average axial force over time, front turbine.

(b) Average axial force over time, rear turbine.

(c) Average azimuthal force over time, front turbine.

(d) Average azimuthal force over time, rear turbine.

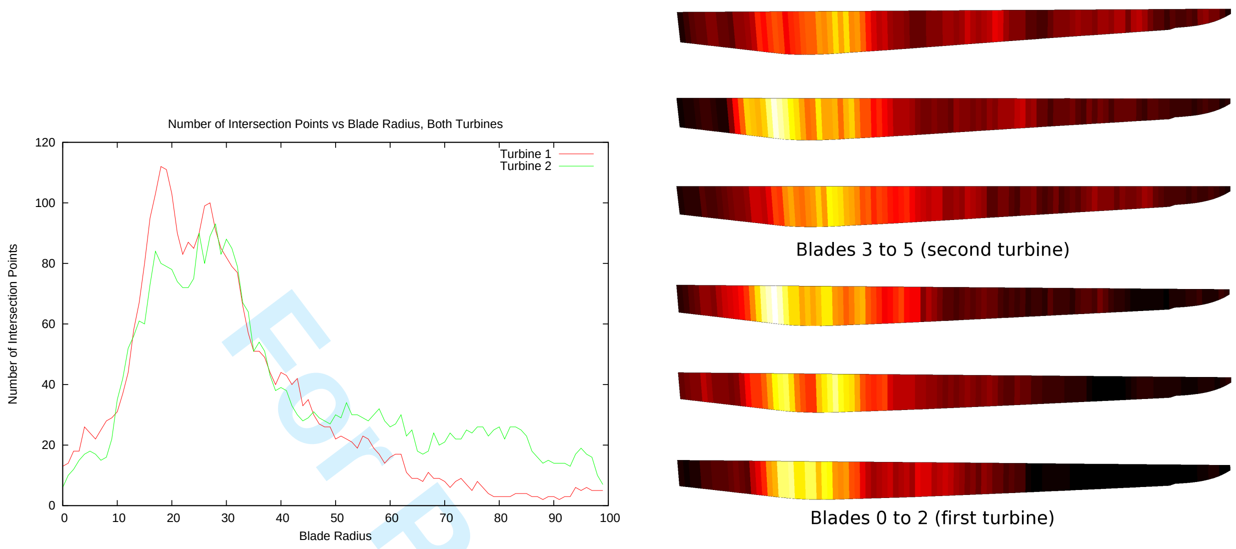
(e) Visualization of vortices, before they reach second turbine.

(f) Visualization of vortices, after they reach second turbine.

Fig. 12. Average axial force versus time for the front and rear turbines (12(a) and 12(b)) and average azimuthal force versus time (12(c) and 12(d)). For each turbine, the red, green and blue lines correspond to the graphs of the first, second and third blades, respectively. We show a visualization of the vortices before they reach the downstream turbine in 12(e). The range of force values for the rear turbine becomes larger and oscillate compared to the first turbine's force values around time step ninety, and this event corresponds to the approaching wake extracted by our visualization system in Figure 12(f). Visual correlation of such events in the global real-space, static blades and statistics views allows for effortless analysis between force and vortex wake behaviors.

farm. It visualizes turbulent flow behavior between turbines in physical space, and allows the viewer to see intricate vortex-blade intersection configurations in a static blades view. Furthermore, our computed statistical data of such intersections make a quantitative scientific analysis possible. While real turbines contain sensors on each blade that measure stress, the respective sensor-generated data has not been analyzed in relation to the environment that the turbines exist in. There has been some work to model turbines on a smaller scale in wind tunnels; however, it is difficult to scale both the aerodynamic and material properties in order to produce data that is representative of real-world turbines.

While our visualization pipeline allows a domain scientist to study a complex time-varying wind turbine simulation, one could extend our system in many ways. We plan to investigate the relationship between power-output and the intersections further since our implementation indicates a strong correlation between the two. Since our intersection-force histograms hint at an interesting trend, one would need to investigate ways to make use of those statistics since intersections in areas of high force values can be related to



(a) Number of intersections versus blade radius for both turbines. Red line is first turbine; green is second.

(b) Colormap of intersection count statistic in static blades view. Darker colors indicate less intersections; brighter colors indicate more.

Fig. 13. Graph representing the number of intersections as a function of blade radius (for each turbine) in 13(a), summed over all time steps. For each turbine, the intersections counts for all three blades are summed together per quadrilateral. The related static blades view is shown in 13(b). The red and green lines correspond to the graphs of the first and second turbines, respectively. The second turbine has a greater number of intersections closer to the tip of the blade.

damaging stress on the turbines. To that end, we hope to apply our methods to a simulation that quantifies damage to the turbine blades and/or rotor system. This would allow us to measure turbine wear and relate it to potential causes.

Acknowledgments

This work was partially supported by the Materials Design Institute, funded by the UC Davis/Los Alamos National Laboratory Educational Research Collaboration (LANL Agreement No. 75782-001-09). It was also supported by the LANL LDRD program under 20100040DR, by the National Science Foundation under contracts IIS 0916289 and IIS 1018097, the Office of Advanced Scientific Computing Research, Office of Science, of the US DOE under Contract No. DE-FC02-06ER25780 through the SciDAC programs VACET, and contract DE-FC02-12ER26072, SDAV Institute. LANL Institutional Computing provided computational resources for the numerical simulations for the data set used here. We thank our colleagues from LANL, Institute for Data Analysis and Visualization, UC Davis, and Simon Stegmaier for his code from [3].

References

[1] H. Lugt, *Vortex flow in nature and technology*. Wiley, 1983.

[2] M. Jiang, R. Machiraju, and D. Thompson, "Detection and visualization of vortices," in *The Visualization Handbook*, C. D. Hansen and C. R. Johnson, Eds. Elsevier, Amsterdam, 2005, pp. 295–309.

[3] S. Stegmaier, U. Rist, and T. Ertl, "Opening the can of worms: An exploration tool for vortical flows," *IEEE Visualization Conference*, pp. 463–470, 2005.

[4] J. Jeong and F. Hussain, "On the identification of a vortex," *Journal of Fluid Mechanics*, vol. 285, pp. 69–94, 1995.

[5] R. Linn and E. Koo, "Determining effects of turbine blades on fluid motion," Los Alamos National Laboratory, Tech. Rep., 2008, patent pending: submitted 10-20-2008.

[6] J. Reisner, V. Mousseau, A. Wyszogrodzki, and D. Knoll, "An implicitly balanced hurricane model with physics-based preconditioning," *Monthly Weather Review*, vol. 133, no. 4, pp. 1003–22, 2005.

[7] R. Linn, J. Winterkamp, D. Weise, and C. Edminster, "Numerical study of slope and fuel structure effects on coupled wildfire behavior," *International Journal of Wildland Fire*, pp. 179–201, 2010.

[8] F. Pimont, J. Dupuy, R. Linn, , and S. Dupont, "Validation of firetec wind-flows over a canopy and a fuel-break," *International Journal of Wildland Fire*, vol. 18, no. 7, pp. 775–790, 2009.

[9] H. JCR, A. Wray, and P. Moin, "Eddies, stream, and convergence zones in turbulent flows," *Center for Turbulence Research Report CTR-S88*, pp. 193–208, 1988.

[10] U. Dallmann, "Topological structures of three-dimensional vortex flow separation," in *American Institute of Aeronautics and Astronautics, Fluid and Plasma Dynamics Conference, 16 th, Danvers, MA*, 1983.

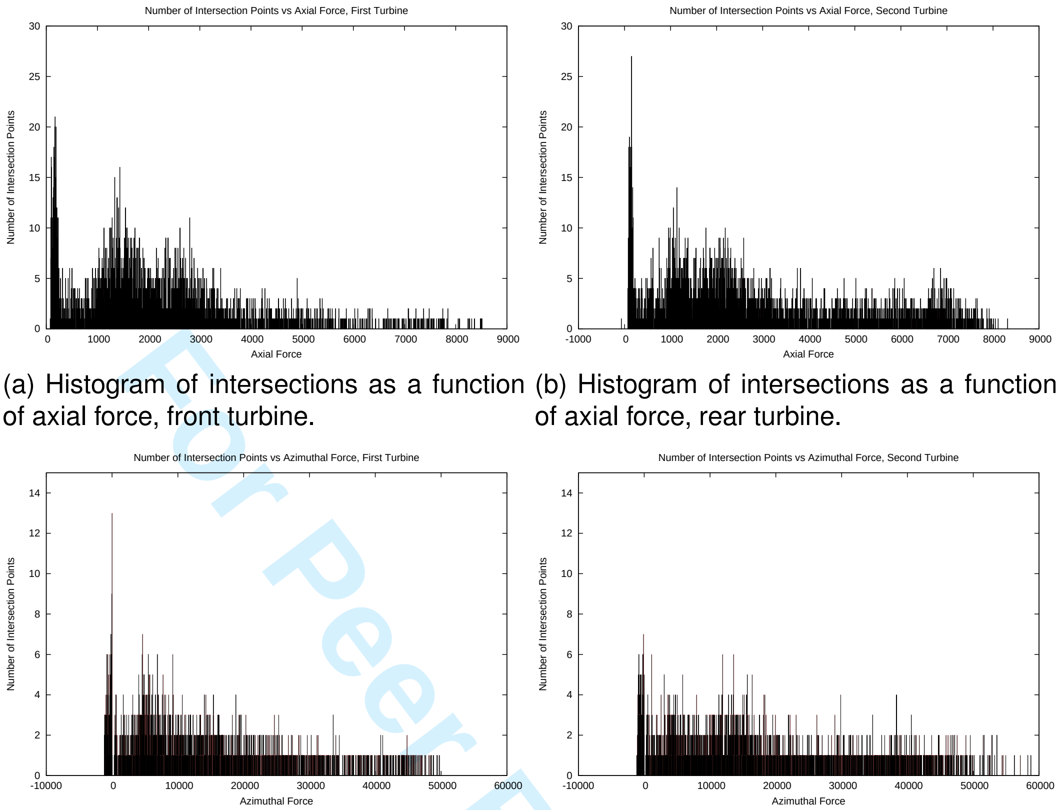
[11] M. Roth and R. Peikert, "A higher-order method for finding vortex core lines," in *Proceedings of the conference on Visualization'98*. IEEE Computer Society Press, 1998, pp. 143–150.

[12] D. Kenwright and R. Haimes, "Automatic vortex core detection," *IEEE Computer Graphics and Applications*, vol. 18, pp. 70–74, 1998.

[13] D. Sujudi and R. Haimes, "Identification of swirling flow in 3D vector fields," in *AIAA 12th Computational Fluid Dynamics Conference, Paper 95-1715*, 1995.

[14] A. Globus, C. Levit, and T. Lasinski, "A tool for visualizing the topology of three-dimensional vector fields," in *Proceedings of the 2nd Conference on Visualization'91*. IEEE Computer Society Press, 1991, pp. 33–40.

[15] C. Garth, X. Tricoche, T. Salzbrunn, and G. Scheuermann, "Surface techniques for vortex visualization," in *Proceedings Eurographics - IEEE TCVG Symposium on Visualization*, May 2004, pp. 155–164.



(a) Histogram of intersections as a function of axial force, front turbine.

(b) Histogram of intersections as a function of axial force, rear turbine.

(c) Histogram of intersections as a function of azimuthal force, front turbine.

(d) Histogram of intersections as a function of azimuthal force, rear turbine.

Fig. 14. Histograms of the number of intersections as a function of axial and azimuthal force, computed for all time steps. Overall, the histograms for the first and second turbines are fairly similar. The second turbine receives slightly more intersections in portions relating to high values of force, and this trend is fairly noticeable if one compares the histograms of axial force. This result indicates that there exists a correlation between high values of force and the vortex intersections.

[16] J. C. R. Hunt, “Vorticity and vortex dynamics in complex turbulent flows,” *Canadian Society for Mechanical Engineering, Transactions*, vol. 11, no. 1, pp. 21–35, 1987.

[17] J. Sahner, T. Weinkauf, and H. Hege, “Galilean invariant extraction and iconic representation of vortex core lines,” in *IEEE VGTC Symposium on Visualization*, 2005, pp. 151–160.

[18] J. Sahner, T. Weinkauf, N. Teuber, and H.-C. Hege, “Vortex and strain skeletons in eulerian and lagrangian frames,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 5, pp. 980 –990, sept.-oct. 2007.

[19] B. Singer and D. Banks, “A predictor-corrector scheme for vortex identification,” *Technical Report: TR-94-11*, 1994.

[20] D. Banks and B. Singer, “Vortex tubes in turbulent flows: identification, representation, reconstruction,” in *Proceedings of the Conference on Visualization’94*. IEEE Computer Society Press, 1994, pp. 132–139.

[21] D. C. Banks and B. A. Singer, “A predictor-corrector technique for visualizing unsteady flow,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, pp. 151–163, 1995.

[22] T. Schafhitzel, D. Weiskopf, and T. Ertl, “Interactive investigation and visualization of 3D vortex structures,” in *Electronic Proceedings of 12th International Symposium on Flow Visualization*, September 2006.

[23] T. Schafhitzel, J. Vollrath, J. Gois, D. Weiskopf, A. Castelo, and T. Ertl, “Topology-preserving λ2-based vortex Core line detection for flow visualization,” in *Computer Graphics Forum*, vol. 27. Wiley Online Library, 2008, pp. 1023–1030.

[24] K. Baysal, T. Schafhitzel, T. Ertl, and U. Rist, “Extraction and visualization of flow features,” in *Imaging Measurement Methods for Flow Analysis*, ser. Notes on Numerical Fluid Mechanics and Multidisciplinary Design, W. Nitsche and C. Dobriloff, Eds. Springer Berlin / Heidelberg, 2009, vol. 106, pp. 305–314.

[25] F. Post, B. Vrolijk, H. Hauser, R. Laramee, and H. Doleisch, “The state of the art in flow visualisation: Feature extraction and tracking,” in *Computer Graphics Forum*, vol. 22, no. 4. Wiley Online Library, 2003, pp. 775–792.

[26] R. Hooke and T. Jeeves, ““Direct Search” Solution of Numerical and Statistical Problems,” *Journal of the ACM (JACM)*, vol. 8, no. 2, pp. 212–229, 1961.

[27] J. F. Blinn, “A generalization of algebraic surface drawing,” *ACM Trans. Graph.*, vol. 1, no. 3, pp. 235–256, Jul. 1982.

[28] M. Brill, H. Hagen, H.-C. Rodrian, W. Djatschin, and S. Klimenko, “Streamball techniques for flow visualization,” in *Visualization, 1994., Visualization ’94, Proceedings., IEEE Conference on*, oct 1994, pp. 225 –231, CP25.

[29] F. C. Crow, “Shadow algorithms for computer graphics,” in *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH ’77, 1977, pp. 242–248.

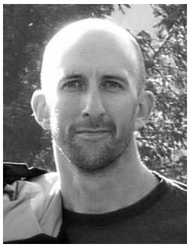
[30] B. Sanderse, “Aerodynamics of wind turbine wakes,” *Energy Research Center of the Netherlands (ECN), ECN-E–09-016, Petten, The Netherlands, Tech. Rep*, 2009.

[31] T. Talay, *Introduction to the Aerodynamics of Flight*. Scientific and Technical Information Office, National Aeronautics and Space Administration, 1975, vol. 367.

[32] N. Kelley, M. Shirazi, D. Jager, S. Wilde, J. Adams, M. Buhl, P. Sullivan, and E. Patton, *Lamar low-level jet project interim report*. National Renewable Energy Laboratory, 2004.

Sohail Shafii received his bachelors degree from University of California, Davis in 2006. He is currently a Computer Science PhD candidate at University of California, Davis, working for the Institute for Data Analysis and Visualization (IDAV) and Los Alamos National Laboratory (LANL). His research background consists of visualization in relation to LiDAR (Light Detection and Ranging), topology, computational fluid dynamics, and data compression.

Harald Obermaier received the master's and PhD (Dr. rer. nat.) degrees in computerscience at the University of Kaiserslautern, Germany in 2008 and 2011. He is currently a postdoctoral researcher at the Institute for Data Analysis and Visualization of the University of California Davis. His research interests lie in scientific visualization and data analysis with applications in areas such as continuum mechanics and geophysics.

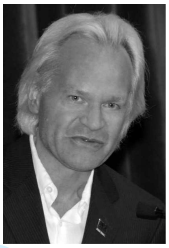


Rodman Linn is a scientist and team leader in the Earth and Environmental Sciences Division at the Los Alamos National Laboratory (LANL), U.S.A. Rodman's work at LANL is predominantly focused on computational modeling of coupled atmospheric phenomena including wind energy, wildfires, urban conflagrations and ecosystem/atmosphere interaction. He received his Ph.D. in Mechanical Engineering from the New Mexico Sate University in 1997.

Eunmo Koo is a scientist at Earth and Environmental Sciences Division at the Los Alamos National Laboratory, U.S.A. His works at the LANL are focused on computational modeling of atmospheric phenomena including wind energy, wildfires and urban conflagrations. He received his Ph.D. in Mechanical Engineering from the University of California, Berkeley in 2006.

Mario Hlawitchska studied computer sciences and electrical engineering with a focus on signal processing and visualization, and received his B.Sc. and M.Sc. (Diplom-Informatiker) degrees from the University of Kaiserslautern, Germany, in 2004. He received a Ph.d. (Dr. rer. nat.) from the Universität Leipzig, Germany in 2008 for his work on visualization of medical data, and completed post-doctoral research in the Department of Biomedical Engineering and the Department of Computer Science at the University of California, Davis. Dr. Hlawitschka is currently a professor for scientific visualization at the Department of Computer Science at the Universität Leipzig.

Christoph Garth received the PhD (Dr. rer. nat.) degree in computer science from the University of Kaiserslautern in 2007 and spent his postdoctoral time as a researcher with the Institute for Data Analysis and Visualization at the University of California, Davis. He is currently an assistant professor of computer science at the University of Kaiserslautern. His research interests include scientific visualization, analysis of vector and tensor fields, topological methods, query-driven visualization, and parallel/scalable algorithms for visualization. He is a member of the IEEE.



Bernd Hamann studied computer science and mathematics at the Technical University of Braunschweig, Germany, and computer science at Arizona State University. At the University of California, Davis, his teaching and research areas are visualization, geometric modeling and computer graphics.

Kenneth I Joy is a Professor in the Department of Computer Science and Director of the Institute for Data Analysis and Visualization at UC Davis. He came to UC Davis in 1980 in the Department of Mathematics and was a founding member of the Computer Science Department in 1983. Professor Joy is a faculty computer scientist at Lawrence Berkeley National Laboratory and is a participating guest researcher at Lawrence Livermore National Laboratory. His areas of interest lie in the fields of visualization, geometric modeling, and computer graphics, where he leads research efforts in multiresolution representations of large scale data sets, visualization of multidimensional data, applications and visualization algorithms to imaging problems, and simplification of data sets resulting from terascale simulations. Professor Joy received a B.A. (1968) and M.A. (1972) in Mathematics from UCLA, and a Ph.D. (1976) from the University of Colorado, Boulder. He is a member of the Association for Computing Machinery (ACM), the IEEE.