Vortex Visualization in Ultra Low Reynolds Number Insect Flight

Christopher Koehler, Thomas Wischgoll, Haibo Dong, and Zachary Gaston

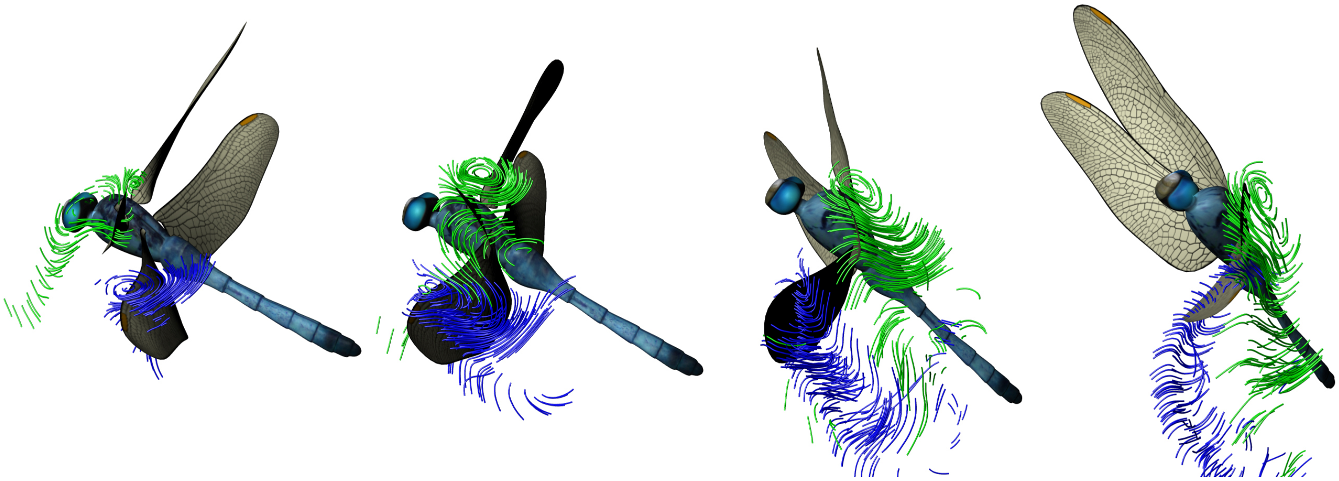

Fig. 1. Vortex formation, attachment and shedding on the left forewing and hindwing of a dragonfly during takeoff.

Abstract—We present the visual analysis of a biologically inspired CFD simulation of the deformable flapping wings of a dragonfly as it takes off and begins to maneuver, using vortex detection and integration-based flow lines. The additional seed placement and perceptual challenges introduced by having multiple dynamically deforming objects in the highly unsteady 3D flow domain are addressed. A brief overview of the high speed photogrammetry setup used to capture the dragonfly takeoff, parametric surfaces used for wing reconstruction, CFD solver and underlying flapping flight theory is presented to clarify the importance of several unsteady flight mechanisms, such as the leading edge vortex, that are captured visually. A novel interactive seed placement method is used to simplify the generation of seed curves that stay in the vicinity of relevant flow phenomena as they move with the flapping wings. This method allows a user to define and evaluate the quality of a seed's trajectory over time while working with a single time step. The seed curves are then used to place particles, streamlines and generalized streak lines. The novel concept of flowing seeds is also introduced in order to add visual context about the instantaneous vector fields surrounding smoothly animate streak lines. Tests show this method to be particularly effective at visually capturing vortices that move quickly or that exist for a very brief period of time. In addition, an automatic camera animation method is used to address occlusion issues caused when animating the immersed wing boundaries alongside many geometric flow lines. Each visualization method is presented at multiple time steps during the up-stroke and down-stroke to highlight the formation, attachment and shedding of the leading edge vortices in pairs of wings. Also, the visualizations show evidence of wake capture at stroke reversal which suggests the existence of previously unknown unsteady lift generation mechanisms that are unique to quad wing insects.

Index Terms—Flow visualization, flowing seed points, streak lines, streamlines, insect flight, vortex visualization, unsteady flow.

1 Introduction

Insects are the undisputed masters of flapping flight in the micro-aerial world, however their flight seems impossible based on traditional notions of aerodynamics. Recently researchers have been using computational fluid dynamics (CFD) to analyze insect flight [1] in hopes of identifying unsteady lift production mechanisms which could in turn be applied to developing more efficient micro air vehicles (MAVs). In particular, the dragonfly has emerged as an important model organism for the development of quad wing MAVs.

In this work, we present a visual analysis of the flow induced by the flapping wings of a digitally reconstructed Eastern Pondhawk *Erythemis Simplicicollis* dragonfly as it performs a backward takeoff maneuver. Vortex formation surrounding the wings is the most desirable phenomena to capture visually due to its role in lift production. Unsteady 3D flow visualization is already very challenging [2], but the addition of multiple dynamically deforming wings to the flow field further complicates the problem. Feature-based methods for vortex region and core detection as well as integration-based flow lines such as streamlines, path lines, streak lines and generalized streak lines [3] were applied to the dragonfly flow data in hopes of capturing various lift production mechanisms.

A novel integration-based technique which we call flowing seed points is introduced to address several of the issues encountered when applying traditional flow lines to the dragonfly data. Also, a novel interactive seeding method is introduced. It is based on the fact that the most interesting flow phenomena generally occurs near and moves with the wings. This technique allows users to quickly define and evaluate the quality of a moving seed point over a series of thousands of time steps, while only working at a single time step.

- *Christopher Koehler and Thomas Wischgoll are with Wright State University College of Engineering and Computer Science, E-Mail: ckoehler.11@wright.edu, thomas.wischgoll@wright.edu.*
- *Haibo Dong and Zachary Gaston are with Wright State University College of Mechanical and Materials Engineering, E-Mail: haibodong@gmail.com, zachary.gaston@gmail.com.*

Manuscript received 31 March 2011; accepted 1 August 2011; posted online 23 October 2011; mailed on 14 October 2011.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

The remainder of the paper is organized as follows. Section 2 contains a discussion of related works. Section 3 explains the photogrammetry system used to reconstruct the flapping dragonfly wings. Section 4 reviews the visualization techniques that were applied, introduces the flowing seed point method and explains our seed curve generation algorithm. Results of all these techniques are shown in section 5 at several portions of a wing beat. Section 6 presents a domain expert's interpretation of the visualizations and their relationship to lift generation, and section 7 reviews the main points as well as our plans for future work in this area.

2 Related Work

This section discusses previous work related to unsteady flow visualization of insect flight. Flapping flight theory is covered, so that the discussion of visualization methods can be done with a firm understanding of which flow features are important in the production of lift. Related works in the areas of unsteady flow visualization, vortex detection and applied flow visualization are also presented. For an overview of the general issues of flow visualization and vortex detection please refer to [2, 4].

2.1 Flapping Flight Theory

Insects operate at ultra low Reynolds numbers where viscous effects are dominant. Several theories as to how they generate the necessary lift and propulsion have been proposed. For dragonfly flight, the most relevant theory is the leading edge vortex hypothesis [5]. The theory is that fluid separates from the sharp leading edge of an airfoil and is then drawn into a vortex, which creates a low pressure region above the wing resulting in higher lift production.

Studies of the leading edge vortex (LEV) have produced different results in 2D and 3D. Dickinson and Götz showed that in a 2D air foil moving with a high angle of attack, the LEV grows until flow reattachment can no longer occur [6]. Sane demonstrated that lift production in 2D air foils is time dependent with the maximum lift occurring just before the LEV is shed [7]. In a 3D study Ellington et al. observed that the LEV grows to a certain point and then remains attached during the down-stroke [8]. Willmott's work with tethered hawkmoths [9] has also identified the dynamic stall that occurs when the angle of attack is rapidly changed as a way to generate lift. In addition Dickinson et al. concluded that delayed stall, rotational circulation and wake capture all contribute to lift production [10].

Although researchers in the bio-fluid field [11] agree that the LEV is the main source of lift production for large insects, it is not clear why it grows to a certain size and then stabilizes in the 3D case but not the 2D case. Thus, the primary goal of any visualization method applied to this data is to capture the formation, attachment and shedding of leading edge vortices as well as clues, such as spanwise flow, as to why they are more stable in the 3D case. A secondary visualization goal is to capture visual evidence of unsteady lift production mechanisms unique to quad wing insects, such as wake capture between forewing and hindwing.

2.2 Unsteady Flow Visualization

Jobard and Lefer visualized 2D unsteady flow with animated streamlines [12]. Wiebel et al. introduced the concept of generalized streak lines, which are streak lines with moving seed points [3]. Wiebel and Scheuermann visualized 4D flow by generating surfaces from bundles of path lines that pass through a common eyelet point and different times [13]. Helgeland and Elboth presented a texture-based method where evenly spaced particles are advected through an unsteady 3D flow [14]. At each time step, field lines are generated at each particle. Cuntz et al. used time-adaptive streamlines, path lines and streak lines to visualize a typhoon data set [15]. More recently, Weinkauf and Theisel developed a method for deriving a streak line vector field in order to facilitate a mathematical analysis of streak lines [16].

Recently, much of the work in unsteady 3D flow visualization has focused on integral surfaces. Garth et al. developed an algorithm for generating integral surfaces that is based on successive timeline approximation [17]. Von Funck et al. presented a method for interactively generating streak surfaces by skipping the typical adaptive re-meshing step and hiding the artifacts with smoke rendering [18]. McLoughlin et al. introduced an algorithm for generating stream and path surfaces that is fast and simple enough to use in practical applications [19]. Krishnan et al. proposed an efficient method of generating time and streak surfaces in large unsteady 3D vector fields [20]. Born et al. used illustrative rendering techniques to address the self occlusion problems of stream surfaces [21]. Ferstl et al. presented a method for visually analyzing separating streak surfaces in unsteady 3D flows [22].

2.3 Vortex Region and Core Detection

Several previous works have addressed detection of vortex cores. Sujudi and Haimes developed a core line detection method based on critical point theory that uses eigenvalues and eigenvectors of the velocity gradient [23]. Roth and Peikert presented a higher-order method for vortex core extraction [24]. Sahner et al. proposed a method of detecting Galilean invariant vortex cores by extracting ridge lines in the output vortex region detection algorithms [25]. Weinkauf et al. generalized the method of Sujidi and Haimes to path lines in order to identify cores of swirling particles [26]. Jiang et al. used geometric properties of streamlines surrounding candidate vortices to verify the output of vortex detection algorithms [27].

Vortex region detection has also been addressed in the literature. Jeong and Hussain proposed a Galilean invariant method, known as the λ_2-criterion [28]. Jiang et al. presented a method for identifying regions of swirling flow by using combinatorial topology [29]. Haller introduced a vortex region criterion that is invariant under general frame of reference changes such as rotations [30].

Others have proposed enhancements to vortex detection. Stegmaier et al. developed a tool for separating and browsing $\lambda_2$vortices in turbulent flows [31]. Tricoche et al. used vector field topology projected onto a cutting plane rotating around a recirculation bubble axis to capture vortex breakdown [32]. Peikert and Sadlo presented topological methods for detecting vortex breakdown bubbles and vortex rings [33]. Jankun-Kelly et al. developed a feature based vortex hull extraction method capable of handling merging vortices [34]. Sadlo et al. visualized vorticity transport along pathlines passing through vortex regions [35].

2.4 Applied Visualization

There have also been several previous works applying cutting edge flow visualization techniques to real world inspired applications. Laramee et al. presented a visual exploration of fluid flow through a cooling jacket [36] using direct, geometric, texture-based and feature-based methods. Bauer et al. presented a case study of unsteady flow visualization where the rotating helix structure, known as a vortex rope, which builds in the draft tube of a wind turbine was captured [37]. You et al. studied vortex shedding and motion in a flow containing a flexible plate using a series of iso-line visualizations [38]. In an application similar to insect flight, Pivkin et al. used streamlines, path lines and particles to visualize air flow around a bat's wings [39].

3 Background

In order to be confident in one's visualization results it is essential to start with accurate data. Thus, this section gives a brief explanation of the processes involved with generating an accurate simulation of dragonfly flight.

3.1 Photogrammetry

The photogrammetry setup used to capture images of dragonflies in flight consists of three synchronized Photron FASTCAM SA3 60K high-speed cameras with 1024x1024 pixel resolution. They were aligned orthogonal to each other and operated at 1000Hz. A calibration object was made using a 3D Systems ProJet HD-3000

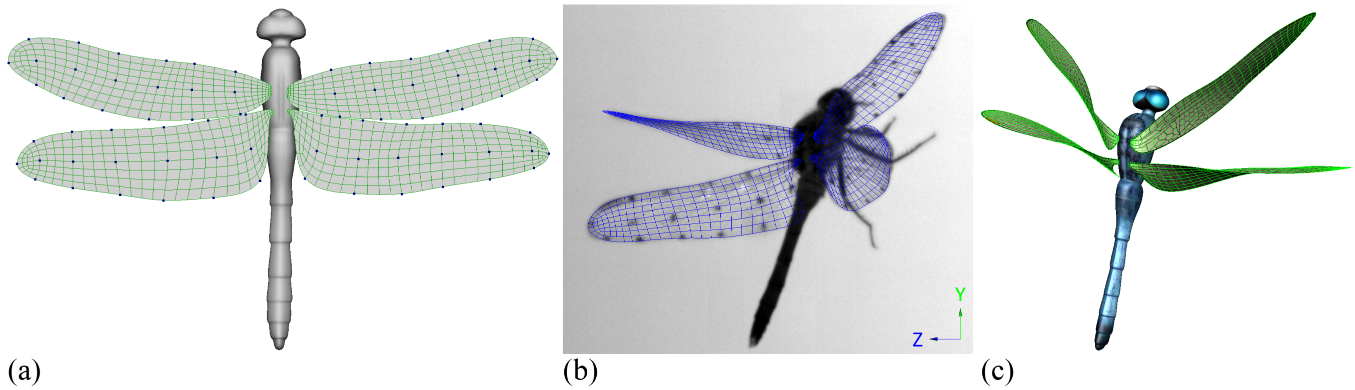

Fig. 2. Dragonfly reconstruction process: (a) Initial wing template surfaces. (b) Alignment of wing templates with reprojected marker points. (c) Wing and body textures added to improve perception of camber and spanwise twist as the wings flap.

rapid prototyping machine and the cameras were calibrated using the direct linear transform algorithm.

3.2 3D Reconstruction

Prior to filming, the dragonfly's wings were each marked in a grid pattern with black ink marker points. These marker points were then tracked in each series of images from the high-speed cameras. Using perspective projection equations and the transformation parameters achieved through calibration, it is possible to reconstruct the true 3D location of marker points found in two or more images for any time step. Template wing models, created with Catmull-Clark subdivision surfaces, were used as the basic shape of each wing (Figure 2a). The 3D reprojected marker points were then converted into surfaces by iteratively aligning the corresponding points on the template models to them (Figure 2b). A texture was applied to the models in order to aid in the perception of wing twist and camber (Figure 2c).

3.3 Simulation

A high-fidelity in-house direct numerical simulation (DNS) tool, which is based on an existing Navier-Stokes immersed-boundary solver [40], was used to simulate the flow induced by the flapping wings. It is capable of simulating flows with complex moving boundaries including membranous and solid bodies on stationary Cartesian grids [41]. The solver employs a non-dissipative, 2nd-order, central-difference scheme [40], which is critical for accurately predicting vortices. Our dragonfly was flying at a Reynolds number of 350 based on its body's reference velocity. The simulation was performed on a Cartesian grid with dimensions 176x152x192. In total, 0.08 seconds of flight, containing three wing beats, was simulated. Vector field data was exported at 1600 time steps, resulting in approximately 1.4 terabytes of data to be visualized.

4 Method

This section discusses the theory behind the visualization techniques applied to the dragonfly data. The flowing seed point method, which combines streamlines and streak lines, is introduced. Also a novel way to interactively define and evaluate seed curves through every time step of an unsteady flow with immersed boundaries is explained.

4.1 Feature-Based Vortex Detection

Vortex feature extraction in the dragonfly simulation was done with the λ_2 vortex region detection algorithm [28] and the Sujudi Haimes eigenvector method [23]. According to the λ_2-definition, the velocity gradient $\boldsymbol{\nabla v}$ is decomposed into its symmetric part, the rate-of-strain tensor $\boldsymbol{S}$, and its antisymmetric part, the vorticity tensor Ω. They then define a vortex as a contiguous region where the intermediate eigenvalue λ_2 of the symmetric real tensor $\boldsymbol{S}^2 + \Omega^2$ is negative. Lower values of λ_2 indicate stronger vortices.

$$\boldsymbol{S} = {}^1\!/_2\,[\boldsymbol{\nabla v} + (\boldsymbol{\nabla v})^{\mathrm{T}}] \qquad \Omega = {}^1\!/_2\,[\boldsymbol{\nabla v} - (\boldsymbol{\nabla v})^{\mathrm{T}}] \tag{1}$$

The eigenvector method for core line extraction [23] identifies locally swirling flow by looking at points where the velocity gradient has one real eigenvalue and a pair of complex-conjugate eigenvalues. At these points, the velocity component in the direction of the eigenvector corresponding to the real eigenvalue is subtracted. Centers of rotation are computed on tetrahedral cell faces and connected with line segments to produce a vortex core lines. Haimes and Kenwright's alignment algorithm [42] was also employed to produce more contiguous core lines.

4.2 Flow Line Theory

Streamlines are trajectories in time independent vector fields. They are tangent to the vector field at all points and are described by the following differential equation:

$$\frac{d\boldsymbol{S}(u)}{du} = \boldsymbol{v}(\boldsymbol{S}(u), t) \tag{2}$$

Where $\boldsymbol{S}(u)$ is a point along the streamline, $\boldsymbol{v}$ is the underlying vector field, u is a time-independent parameter and t selects the simulation time step from which all of the vectors are taken.

Streamlines proved effective for visually capturing the LEV due to the easily identifiable helix shape. They can be used with unsteady flows by integrating from the same seed point or a moving seed point at each time step; however this produces spatially incoherent flickering animations due to the fact that small changes in the vector field at adjacent time steps can lead to large changes in streamline trajectory.

Path lines trace the trajectories of massless particles in unsteady flows. These particle traces are Galilean invariant [26] and they form the basis for many other unsteady flow visualization constructs. For example, streak lines are generated by connecting a series of particles that were released into the flow at consecutive time steps from the same seed point.

Generalized streak lines [3] extend this concept by connecting all particles which were released at consecutive time steps from a point moving along a seed curve. Generalized streak lines are potentially more effective at visualizing data sets without inlet flows and data sets where the most important flow features move throughout the flow domain. The dragonfly data has both of these characteristics. A generalized streak line is illustrated at three time steps in the first row of images in Figure 3.

Swirling patterns in unsteady flows can be identified with streak lines; however they can be perceptually misleading. The swirling shape of a streak line that was in the vortex will remain present for several time steps if the vortex dissipates. This is illustrated in the first row of images in figure 3. The data set used to generate this figure is a simulated flow around a single circular membranous disk. The disk undergoes a pitching and heaving motion. The disk center heaves in the y-direction according to:

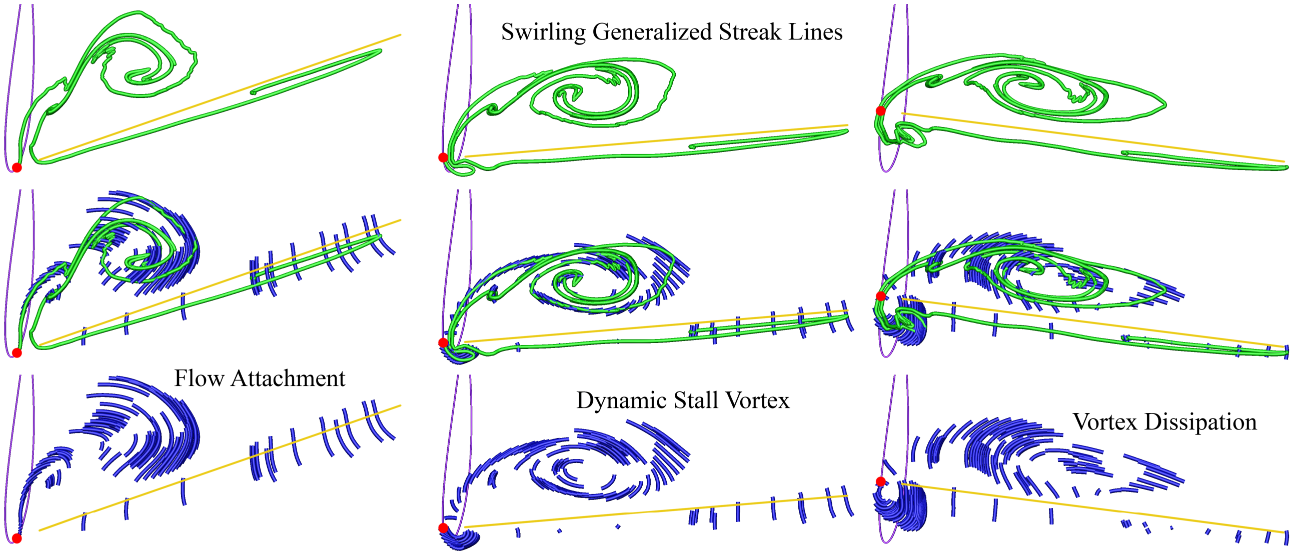

Fig. 3. Comparison of generalized streak lines (green) and flowing seed points (blue) for visualizing a dynamic stall vortex, which briefly appears when the angle of attack of a flapping foil changes rapidly. Each column shows a different time step. The streak line shows a swirling structure at all three time steps, while the flowing seeded streamlets show flow attachment in the first time step and vortex dissipation in the last time step. Particles are emitted from the purple seed curve and the current emission point is marked with a red dot.

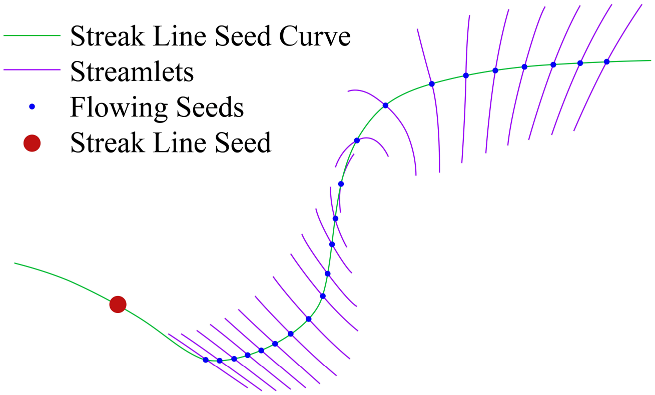

Fig. 4. Illustration of the flowing seed point method. Particles (blue) are emitted from the red seed point and advected through the flow. At each time step, a streamline is computed at each particle location (purple). The corresponding streak line is shown in green.

$$y(t) = A_y \sin(2\pi f t) \quad (3)$$

It pitches about its center according to:

$$\theta(t) = A_\theta \cos(2\pi f t) \quad (4)$$

The heave and pitch amplitudes are A_y and A_θ respectively, θ is the pitch-angle, f is the flapping frequency and t is time. The same formulas were used to generate the generalized streak line seed curve shown in figure 3.

4.3 Flowing Seed Points

Streak lines clearly lend themselves to smooth animation, while streamlines do not. However, streamlines proved more effective at capturing the formation of the leading edge vortex on the dragonfly's wings. Streak lines also suffer when there are flapping immersed boundaries in the flow due to the need for excessive stretching or tearing to prevent intersection as particles separate around the leading edges of the flapping foils. Thus, the purpose of the flowing seed point algorithm is to unify the benefits of each method while still achieving spatially coherent animations.

Conceptually flowing seed points are similar to streak lines, except instead of connecting a series of particles emitted from the same point, a short streamline or streamlet is integrated in the instantaneous vector field at each particle emitted from the same point. As demonstrated by Helgeland and Elboth [14], a shorter integration time for each streamline reduces the spatial incoherence problems streamlines have in unsteady flows. Typical streamlet integration times were 8 to 16 steps in each direction. This concept is illustrated in figure 4 and compared with a generalized streak line in figure 3. Flowing seed points have demonstrated the following benefits when visualizing flapping wing data sets:

- Improved visualization of vortex formation and shedding
- Temporally smooth animation of streamline trajectories
- Visualization of instantaneous and temporal divergence
- Improved handling of immersed boundaries in the flow
- Visualization of instantaneous velocity when the number and size of integration steps is constant for each streamlet

4.4 Seed Curve Generation

Seed point placement is another very challenging aspect of using integration-based flow line visualization with highly unsteady 3D flows. If seeding is too dense visualizations become very busy and suffer from self occlusion. On the other hand, it is more difficult to capture all features of interest with fewer seed points. The effectiveness of all the methods in the previous section relies on having good seed placement for both streamlines and particles.

Seed placement has been addressed in the literature. Verma et al. [43] presented a non-iterative seeding strategy for placing streamline seeds near critical points in 2D steady flows. Ye. et al. used seeding templates in the vicinity of critical points to seed streamlines in 3D flows [44]. Yuan et al. used a similarity measure to grow streamlines from a dense set of potential seeds [45]. Li and Shen proposed an image-based method for streamline seeding in 3D vector fields [46]. Marchesin et al. used a combination of view dependent and streamline metrics for streamline seed placement in 3D flows [47].

Automatic seeding methods based on some vector field metric calculated at each grid point or based on measurements taken from densely seeded flow lines can be very slow when dealing with tera-scale or larger data sets due to IO. It is even more challenging to use such methods to define seed points that move over time because of the constraint that seeds should move gradually between time steps. Also, traditional interactive seed placement methods typically only allow users to place static seed points, curves and surfaces. Static seed points suffer in situations where there is no inlet flow and when there are immersed boundaries in the flow field.

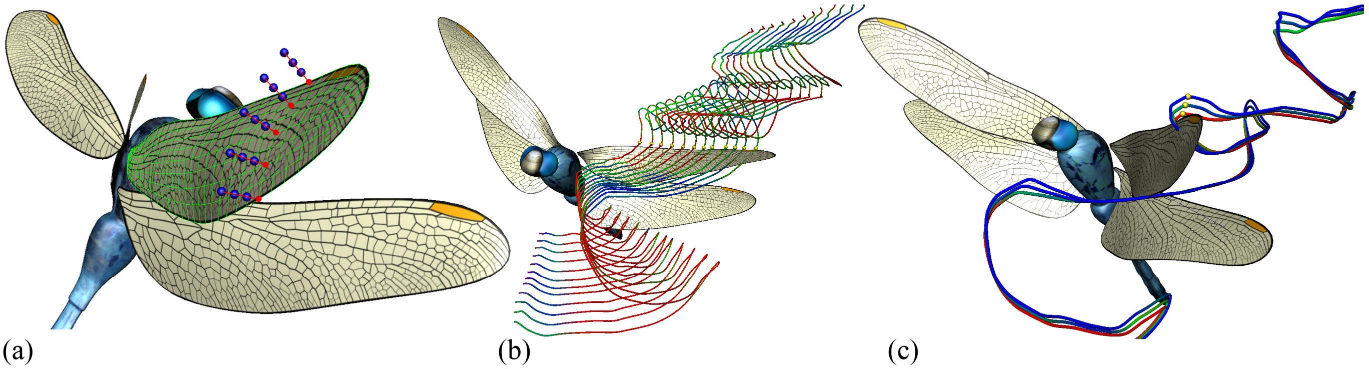

Fig. 5. Immersed wing boundaries aid in generating seed curves over time. Vorticity magnitude was mapped to color at each time step of the seed curve in order to evaluate the quality of the seeds. (a) Seeds are placed a user defined distance from the wing boundary in the normal direction. (b) A series of seed curves normal to the leading edge of the left forewing. (c) Three seed curves placed in the chord direction.

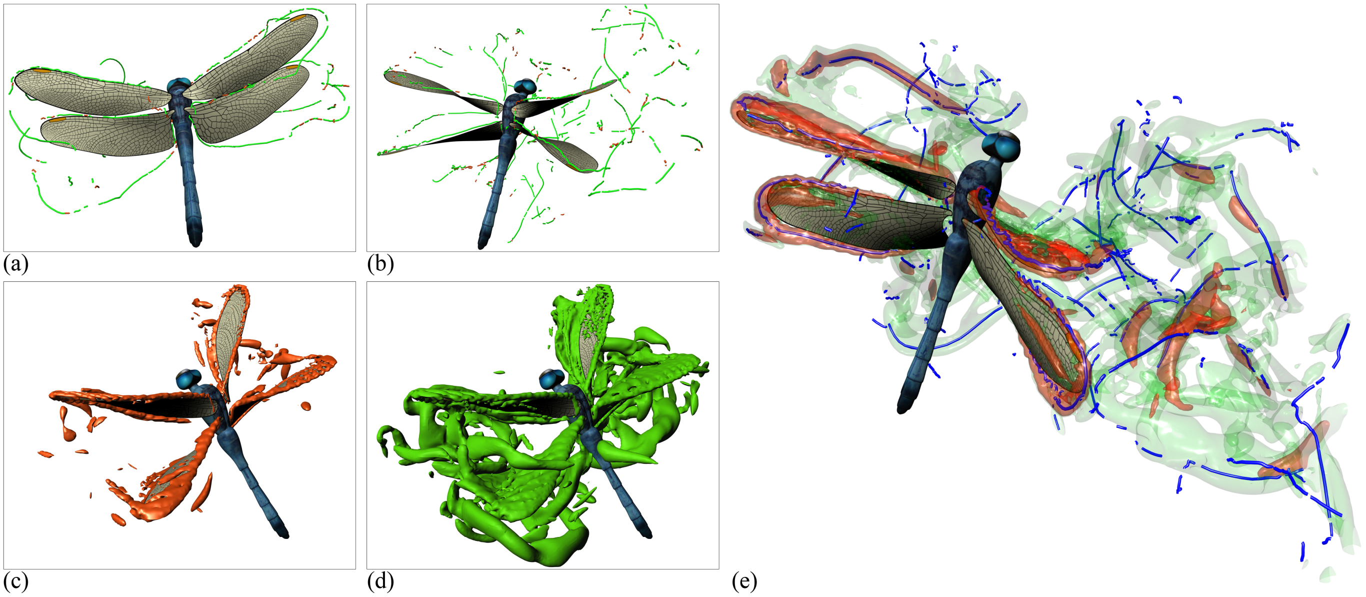

Fig. 6. Feature-based vortex detection methods applied to the dragonfly data. (a)(b) Eigenvector method for core line extraction [23]. (c)(d) Vortex region detection with the λ_2-criterion. (e) Comparison between the region and core line detection results.

Our approach is different in that it uses the immersed boundaries in the flow (in this case the wings) in order to define moving seed points over time. We initially ignore the underlying vector field and assume that the objects moving in the flow are causing the most interesting flow phenomena. Seeds are bound near the surface of the immersed objects in order to capture the neighboring near-field flow features. Specifically, vertex normals at user selected points on objects in the flow domain are used to define the distance of the seed from an immersed boundary (Figure 5a).

In order to use vertex normal seeds effectively, a user must be able to quickly identify surface points in the areas where seeds are desired. To accomplish this, several 3D polygon modeling operations were incorporated into the software. In particular, edge loop selection and surface patch selection are used to define where on the wings seeds should be placed. Subdivision refinement controls the seed density and the mouse scroll wheel controls how far in the normal direction the seeds are placed.

As seed points are defined at one time step, the seed curves are automatically displayed. Seed curves are obtained by connecting points at neighboring time steps that are the same distance in the normal direction off the same point on the wing mesh. Scalar quantities of the underlying vector field, such as velocity, vorticity or λ_2 are then mapped to the color of each seed curve. This allows the user to interactively evaluate whether a seed curve will capture an interesting portion of the flow at enough time steps before computing any flow lines. This is shown in figure 5b and 5c. This seed point placement method has the following advantages over seeding methods that use vector field metrics as well as traditional interactive methods such as static rakes near an inlet flow:

- Speed of seed placement is unaffected by vector field size
- Intuitive interactive seed curve creation and evaluation
- Improved temporal and spatial coherency in animations

5 Visualizations

This section presents the results of using the aforementioned visualization methods to study air flow around the reconstructed dragonfly wings. The flow phenomena is very similar on both sides of the insect, so certain methods were applied only to wings on the same side in order to reduce occlusion and information overload.

5.1 Vortex Detection

The λ_2 vortex region detection algorithm [28] and the Eigenvector method for core line extraction [23] were applied to the dragonfly data. Both methods proved effective at capturing the existence of the LEV (figure 6a and 6c). In general vortex core and region detection algorithms performed better in the early time steps, before a complex wake structure had time to form (figure 6a). At the later time steps core line detection suffered from scale dependence issues where the strong LEV was lost in the cores of all the weak vortices interacting in the downwash (figure 6b and 6e). Choosing an appropriate threshold

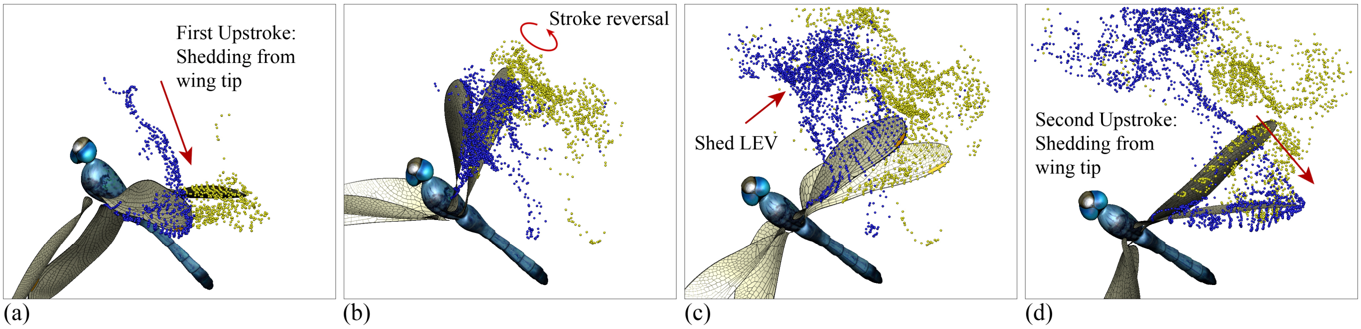
First Upstroke: Shedding from wing tip
Stroke reversal
Shed LEV
Second Upstroke: Shedding from wing tip
(a)
(b)
(c)
(d)

Fig. 7. Particles continuously emitted near the leading edges of the right pair of wings. Blue particles correspond to the forewing and yellow particles correspond to the hindwing. Particles are mainly being shed from the wing tip during the up-stroke. Also, vortex shedding is very apparent during stroke reversal, however it is difficult to identify individual vortices with this visualization method.

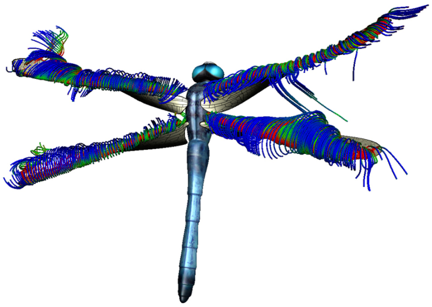

Fig. 8. Streamlines used to visualize the LEV on each wing during an up-stroke. Color is mapped to vorticity magnitude.

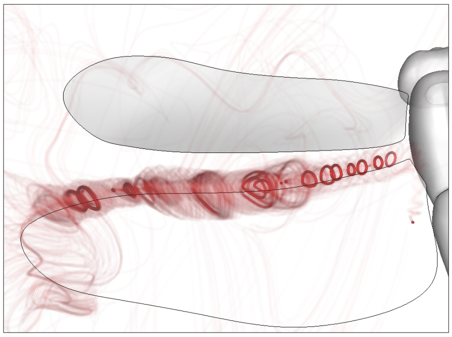

Fig. 9. Semi-transparent streamlines with long integration times used to visualize closed streamlines in left hindwing LEV

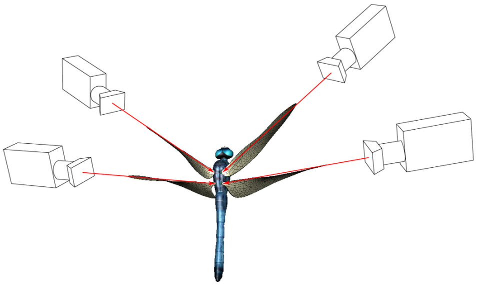

Fig. 10. Cameras bound to the dragonfly wings such that they are looking down the core of the LEV at each time step.

value for λ_2 also proved challenging because different values were needed for different wings and different portions of the stroke. Also, while these methods were able to detect the LEV they provided little insight into why it remains attached to the wings. For these reasons, we chose to complement the vortex detection methods with a series of integration-based methods.

5.2 Particles

Particle advection was tested on the dragonfly data set with varying particle lifetimes and emission rates. Particle emission seeds were bound slightly off the leading edges normal to the wing. Rendering animated particles proved more effective than visualizing the corresponding static path lines due to the fact that they typically intersect the wings. Color was assigned based on which wing is driving the particle emission seeds (Figure 7).

Particles are, for the most part, only being shed off the wing tips when vortices are attached during up-stroke and down-stroke (Figure 7a). Experiments show that vortex shedding can be visually identified using particles by the swirling motions in the flow at stroke reversal (Figure 7c and 7d). However, the drawback of using particles is that it is difficult to identify the tight vortices when they are attached to each wing.

5.3 Streamlines

Streamlines were also tested on the dragonfly data set using vertex normal seeds. We used a fairly dense seeding along the leading edge of each wing so that integration time could be reduced while maintaining good streamline coverage of the entire LEV. The shorter integration time helps reduce flickering as the seed points gradually move in space and time. When animated it is easy to see exactly when in time the LEV on each wing forms due to the easily identifiable helix shape of the streamlines. This is illustrated in figure 8 at a time step when all four wings are performing an up-stroke.

One drawback to using streamlines for visualizing near-field vortex formation in flapping wing insect flight is that it is hard to capture where the vortices go when they are shed at stroke reversal. The seeds are locked a set distance off the leading edges. If the LEV is shed out into the flow, the streamlines will not follow it like particles advected over time would. Thus, we are unable to use streamline visualization alone to look for wake capture.

Closed streamlines can be used to visualize patterns of recirculation in the instantaneous vector fields of unsteady flows. Semi-transparent streamlines with very long integration times were used to identify closed streamlines in the 3D flows because they become darker in areas where the streamlines repeat themselves. Visual evidence of closed streamlines in the vicinity of a leading edge vortex is shown in figure 9. The wings were kept semi-transparent in this example in order to reduce occlusion.

When viewing streamline visualizations of this data set, it is important to note that the wings are represented as membranous objects in the CFD solver. Thus, there are no zero vectors marking their insides in the output. As a wing flaps, if the flow near its surface on both sides is moving in the same direction, then the streamlines will intersect it. This is an artifact of using streamlines on unsteady flow datasets with moving membranous immersed boundaries; however it does not make them less useful as long as users are mindful of the cause.

5.4 Flowing Seed Points

Flowing seeds were also applied to visualize the dragonfly data. The goal of using the flowing seed point method with this data set is to identify if there is any interaction between the flapping wings and previously shed vortices in the flow domain. For this reason, streamlets corresponding to the same wing were given the same color. The emission rate, particle lifetime and integration time of each streamlet were experimentally determined.

Initial tests produced mixed results. At certain viewing angles it was difficult to discern anything that the particle animations did not already tell us. However, at other angles we could clearly track the vortices as they were shed off the leading edge of each wing. This is

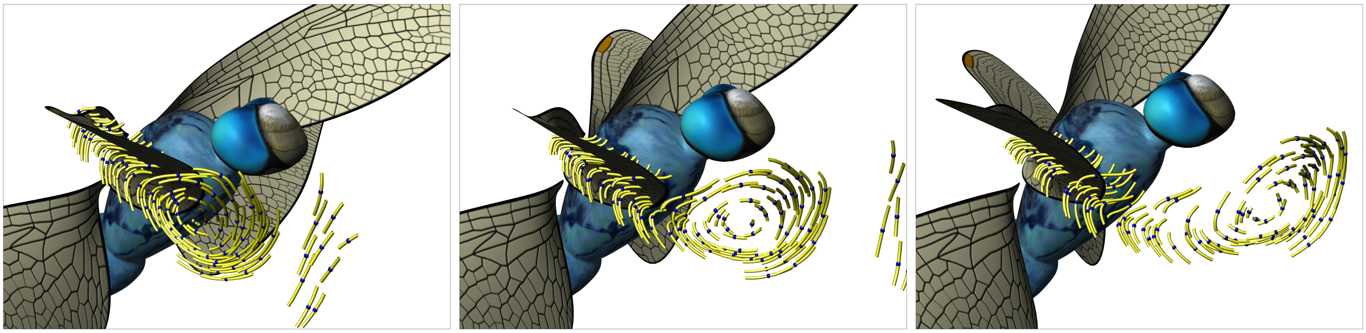

Fig. 11. Flowing seeds emitted from the chord of the right forewing capture vortex shedding during a down-stroke to up-stroke reversal.

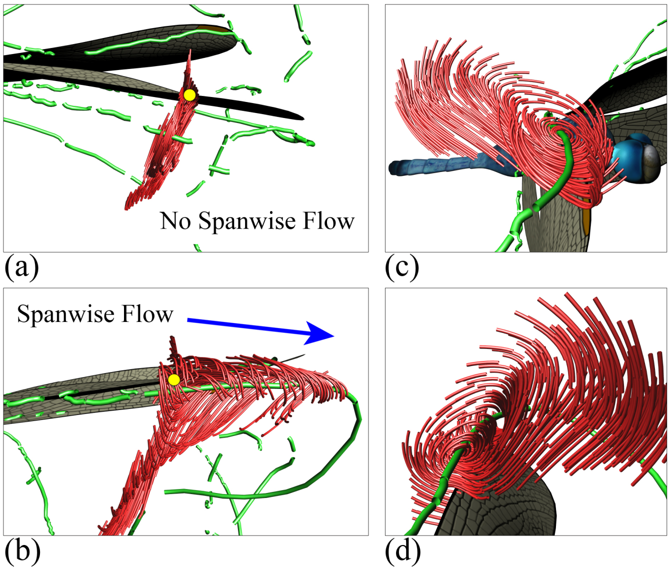

Fig. 12. Spanwise flow along the right forewing LEV at the end of a downstroke: (a) Top down view of flowing seeds (red) injected from a moving emission point (yellow) that remain perpendicular to the leading edge during the first half of the stroke. (b) Flow begins to move in the spanwise direction at the end of the stroke. (c)(d) Comparison between the streamlets and the vortex core (green).

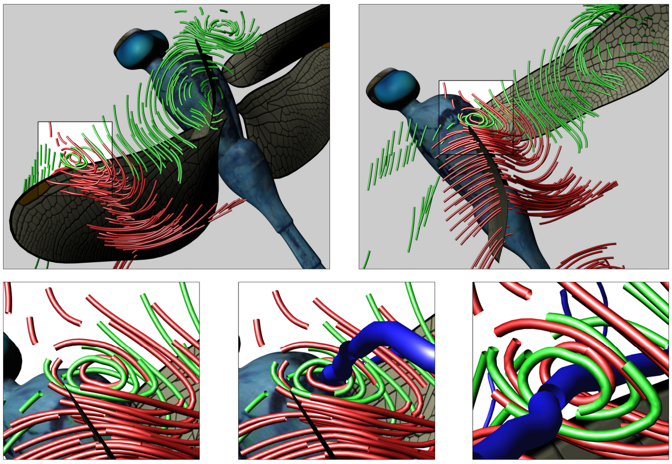

Fig. 13. Visual evidence of wake capture between wings. Flowing seeds injected into the LEV of the forewing during the previous upstroke (green) enter the LEV on the hind wing as it finishes its upstroke. Close up images highlight the vortex on the hindwing and compare it to the extracted core line (blue).

due to the fact that the vortices are moving in a curved figure-eight pattern during each wing beat. In order to fully utilize the benefits of flowing seed points with streamlets it became clear that the viewing angle needed to evolve as the wings flapped.

5.5 Wing Bound Cameras

Our solution is similar to the seed curve placement method in that the immersed wing and body objects are used to drive camera motion. At each time step, a vector from the root to the tip of each wing is saved. Cameras are then placed a user defined distance off the wing tip in the direction of this vector. Camera focal points are fixed at the corresponding wing root. Thus, the view direction of each camera is approximately parallel to the corresponding LEV at all times. Seed curves are then placed densely in the chord direction, instantaneously perpendicular to the viewing angle. Camera alignment is illustrated in figure 10, and the flowing seed point results in figures 1, 11, 12 and 13 utilize wing bound cameras.

Figure 1 uses flowing seeds and a camera mounted to the left forewing to capture the formation and movement of the LEV at four time steps during a down-stroke. Also, vortex shedding was observed at several down-stroke to up-stroke reversals. Figure 11 gives an example of this happening with the right forewing. This visualization includes the particles where each streamlet is seeded. It is clear that the vortex would be indiscernible with particles alone.

Flowing seed points were used to investigate spanwise flow in the LEV. It has been theorized that spanwise flow allows energy to drain out of the LEV, which in turn allows it to remain stable and attached to the 3D wing unlike in the 2D case. The streamlets help identify the vortex while the particle advection captures the flow moving in the spanwise direction. This is illustrated in figure 12. Comparisons were made with vortex core detection to validate the accuracy of the observed vortices and also to illustrate how the spanwise flow along the vortex cannot be identified through the vortex cores alone.

Flowing seed points also captured visual evidence of wake capture between pairs of wings. Vortices are the main source of lift in ultra low Reynolds number insect flight. When the leading edge vortex is shed from a wing, the lift being generated by that wing drops. However, when the vortex is shed from the forewing during the up-stroke to down-stroke reversal, some of its energy appears to be getting recaptured by the hindwing which has already started its down-stroke. Tests need to be done with more specimens and flight modes to say definitively if this is an unsteady lift mechanism unique to the dragonfly, but the visualizations allowed us to identify the possibility. This phenomenon is illustrated in figure 13. Green particles are being injected into the LEV on the forewing and they are clearly entering the LEV of the hindwing soon after being shed.

5.6 Galilean Invariance and Streamlines

Streamlines have proven useful for manually [23, 24] and automatically [27] identifying vortices. However, as pointed out by [26, 31], streamlines will not necessarily form these easily identifiable helix shapes in situations where vortices are moving, due to Galilean invariance. We observed that in the case of strong vortices like the LEV, non-Galilean invariant visualization methods such as streamlines and the Eigenvector method for vortex core detection produce results similar to the Galilean invariant λ_2-criterion. However, these methods produce slightly different results near the

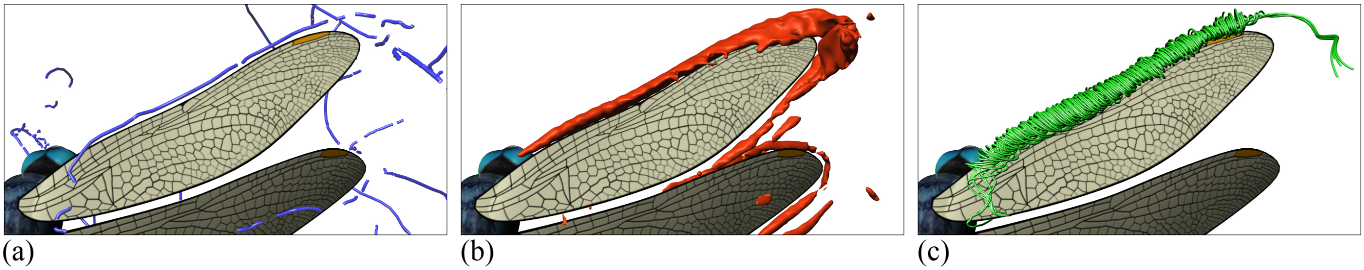

Fig. 14. Comparison between Galilean invariant and non-Galilean invariant methods for capturing the LEV on the right forewing during a down stroke. (a) Eigenvector method for core line extraction [23]. (b) Vortex region detection with the λ_2-criterion [28]. (c) Streamlines seeded slightly off the leading edge.

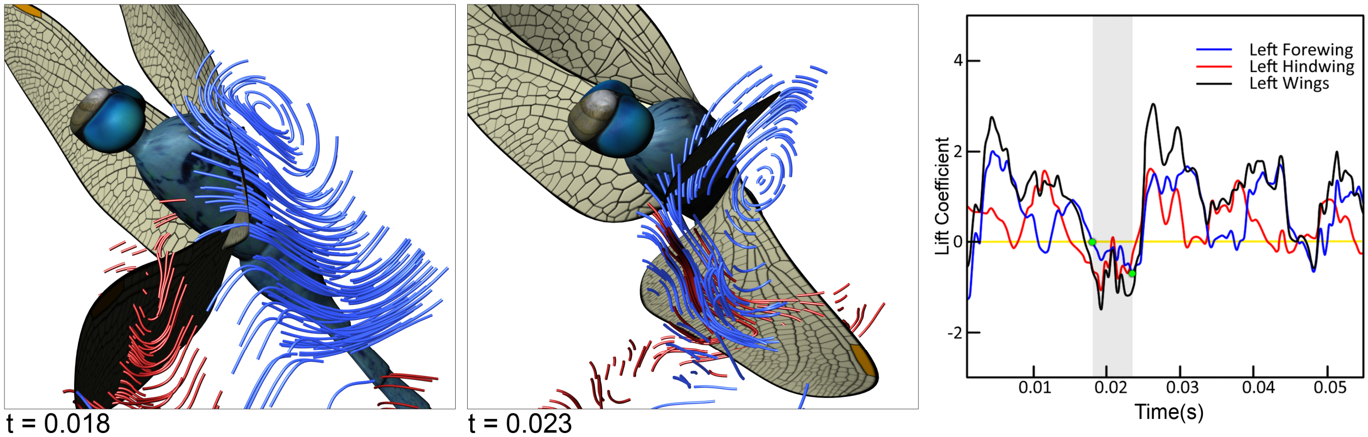

Fig. 15. Flowing seed point visualizations compared to a plot of lift coefficient over time during a down-stroke to up-stroke reversal of the left forewing. The force plot shows that when the vortex is no longer attached to the top of the wing there is a distinct drop in lift coefficient.

wing tips where the vortices are moving fastest and shedding from the wings. This is illustrated in figure 14. Also, we observed differences between these methods when applied to the induced flow, as can be seen in figure 6e.

The flowing seed point method partially addresses the limitations streamlines have due to being non-Galiliean invariant by combining them with particle trajectories that are traced in time. More research is needed to quantify the impact vortex strength and velocity have on streamlines seeded varying distances from vortex cores.

6 Domain Expert Review

In order to gain further insight into the vortex behaviour that we have visualized up to this point, comparisons were made between the visualizations and plots of lift production over time. Peaks and valleys in the lift were identified and visualizations were created at the surrounding time steps in order to make the critical link between vortex formation, attachment and shedding and the resulting force (figure 15). In this figure, the vector fields corresponding to a sharp drop in lift on the left forewing are examined with flowing seed points. The wing is in a down-stroke to up-stroke reversal and the previous LEV is beginning to shed under the wing and out into the flow in front of the dragonfly. As the wing becomes approximately vertical at time 0.018 the lift coefficient is near zero. The ensuing valley in lift production occurs when the shed LEV moves under the wing before the new LEV has formed. This makes sense with our predictions of how the vortices would be behaving when lift drops.

In practice similar comparisons would be made between lift coefficient and all of the visualization methods presented in this paper. We take great satisfaction in the fact that these methods are being adopted by the Wright State University Flow Simulation Research Group for further analysis of different dragonfly flying modes. We hope to use these methods to identify new unsteady lift generation mechanisms in flapping flight and to gain insight into the role of wing deformation in vortex production.

7 Conclusion and Future Work

Clearly insect flight is a very important and challenging application domain for applied flow visualization. Multiple dynamically deforming flapping wings create extremely unsteady flows and it is difficult to predict the effectiveness of any given visualization technique.

We addressed this problem by combining vortex core and region detection methods with a series of integration-based methods. The flowing seed point method was introduced to combine the benefits of streamlines and streak lines in order to visualize vortex shedding, spanwise flow and wake capture between wings. This method also helps clarify some perceptual issues that occur when using streak lines to visualize flapping flight data. We introduced an interactive seed curve generation method that takes advantage of the presence of the flapping wings to greatly simplify seed placement. Also, a camera animation method was demonstrated to keep the focus on the leading edge vortices as they move throughout the flow domain. In the future we plan to extend this work by utilizing a wider range of cutting edge flow visualization techniques and by applying them to a wider array of flapping flight simulations.

Acknowledgments

The authors wish to thank Dr. Hui Wan for helping capture and film the dragonfly used in this work. This project was funded in part by HPTI and the DoD High Performance Computing Modernization Program Office through grant PP-ACE-KY02-013-P3 and DoD DURIP grant FA9550-09-1-0460.

References

[1] W. Shyy, H. Aono, S. K. Chimakurthi, P. Trizila, C. K. Kang, C. E. S. Cesnik and H. Liu, "Recent progress in flapping wing aerodynamics and aeroelasticity," *Prog. Aerospace Sci.*, vol. 46, pp. 284-327, October, 2010.

[2] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post and M. Chen, "Over Two Decades of Integration-Based, Geometric Flow Visualization," *Computer Graphics Forum,* vol. 29, pp. 1807-1829, 2010.

[3] A. Wiebel, X. Tricoche, D. Schneider, H. Janicke and G. Scheuermann, "Generalized Streak Lines: Analysis and Visualization of Boundary Induced Vortices," *IEEE Transactions on Visualization and Computer Graphics,* vol. 13, pp. 1735-1742, 2007.

[4] M. Jiang, R. Machiraju and D. Thompson, "Detection and visualization of vortices," in *The Visualization Handbook*, 2005, pp. 295-309.

[5] E. C. Polhamus, "Predictions of Vortex-Lift Characteristics by a Leading-Edge Suction Analogy," *Journal of Aircraft,* vol. 8, pp. 193-199, April, 1971.

[6] M. H. Dickinson and K. G. Gotz, "Unsteady Aerodynamic Performance of Model Wings at Low Reynolds Numbers," *Journal of Experimental Biology,* vol. 174, pp. 45-64, January, 1993.

[7] S. P. Sane, "The aerodynamics of insect flight," *Journal of Experimental Biology,* vol. 206, pp. 4191-4208, December, 2003.

[8] C. P. Ellington, van den Berg, A. P. Willmott and A. L. R. Thomas, "Leading-edge vortices in insect flight," *Nature,* vol. 384, pp. 626-630, December, 1996.

[9] A. P. Willmott, C. P. Ellington and A. L. R. Thomas, "Flow visualization and unsteady aerodynamics in the Flight of the hawkmoth, Manduca sexta," *Philosophical Transactions: Biological Sciences,* vol. 352, pp. 303-316, March, 1997.

[10] M. H. Dickinson, F. Lehmann and S. P. Sane, "Wing Rotation and the Aerodynamic Basis of Insect Flight," *Science,* vol. 284, pp. 1954-1960, June, 1999.

[11] G. V. Lauder, "Aerodynamics: Flight of the robofly," *Nature,* vol. 412, pp. 688-689, August, 2001.

[12] B. Jobard and W. Lefer, "Unsteady Flow Visualization by Animating Evenly-Spaced Streamlines," *Computer Graphics Forum,* vol. 19, pp. 31-39, 2000.

[13] A. Wiebel and G. Scheuermann, "Eyelet particle tracing - steady visualization of unsteady flow," in *IEEE Visualization '05,* 2005, pp. 607-614.

[14] A. Helgeland and T. Elboth, "High-Quality and Interactive Animations of 3D Time-Varying Vector Fields," *IEEE Transactions on Visualization and Computer Graphics,* vol. 12, pp. 1535-1546, 2006.

[15] N. Cuntz, A. Pritzkau and A. Kolb, "Time-Adaptive Lines for the Interactive Visualization of Unsteady Flow Data Sets," *Computer Graphics Forum,* vol. 28, pp. 2165-2175, 2009.

[16] T. Weinkauf and H. Theisel, "Streak Lines as Tangent Curves of a Derived Vector Field," *IEEE Transactions on Visualization and Computer Graphics,* vol. 16, pp. 1225-1234, 2010.

[17] C. Garth, H. Krishnan, X. Tricoche, T. Bobach and K. I. Joy, "Generation of Accurate Integral Surfaces in Time-Dependent Vector Fields," *IEEE Transactions on Visualization and Computer Graphics,* vol. 14, pp. 1404-1411, 2008.

[18] W. von Funck, T. Weinkauf, H. Theisel and H. Seidel, "Smoke Surfaces: An Interactive Flow Visualization Technique Inspired by Real-World Flow Experiments," *IEEE Transactions on Visualization and Computer Graphics,* vol. 14, pp. 1396-1403, November-December, 2008.

[19] T. McLoughlin, R. S. Laramee and E. Zhang, "Easy integral surfaces: A fast, quad-based stream and path surface algorithm," in *CGI '09: Computer Graphics International,* Victoria, British Columbia, Canada, 2009, pp. 73-82.

[20] H. Krishnan, C. Garth and K. I. Joy, "Time and Streak Surfaces for Flow Visualization in Large Time-Varying Data Sets," *IEEE Transactions on Visualization and Computer Graphics,* vol. 15, pp. 1267-1274, December, 2009.

[21] S. Born, A. Wiebel, J. Friedrich, G. Scheuermann and D. Bartz, "Illustrative Stream Surfaces," *IEEE Transactions on Visualization and Computer Graphics,* vol. 16, pp. 1329-1338, 2010.

[22] F. Ferstl, K. Burger, H. Theisel and R. Westermann, "Interactive Separating Streak Surfaces," *IEEE Transactions on Visualization and Computer Graphics,* vol. 16, pp. 1569-1577, 2010.

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

[24] M. Roth and R. Peikert, "A higher-order method for finding vortex core lines," in *IEEE Visualization '98,* Research Triangle Park, North Carolina, United States, 1998, pp. 143-150.

[25] J. Sahner, T. Weinkauf and H. C. Hege, "Galilean invariant extraction and iconic representation of vortex core lines," in *EuroVis '05,* 2005.

[26] T. Weinkauf, J. Sahner, H. Theisel and H. C. Hege, "Cores of Swirling Particle Motion in Unsteady Flows," *IEEE Transactions on Visualization and Computer Graphics,* vol. 13, pp. 1759-1766, 2007.

[27] M. Jiang, R. Machiraju and D. Thompson, "Geometric verification of swirling features in flow fields," in *IEEE Visualization '02,* 2002, pp. 307-314.

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

[29] M. Jiang, R. Machiraju and D. Thompson, "A novel approach to vortex core region detection," in *Proceedings of the Symposium on Data Visualisation 2002,* Barcelona, Spain, 2002, pp. 217.

[30] G. Haller, "An objective definition of a vortex," *Journal of Fluid Mechanics,* vol. 525, pp. 1-26, 2005.

[31] S. Stegmaier, U. Rist and T. Ertl, "Opening the can of worms: An exploration tool for vortical flows," in *IEEE Visualization '05,* 2005, pp. 463-470.

[32] X. Tricoche, C. Garth, G. Kindlmann, E. Deines, G. Scheuermann, M. Ruetten and C. Hansen, "Visualization of intricate flow structures for vortex breakdown analysis," in *IEEE Visualization '04,* 2004, pp. 187-194.

[33] R. Peikert and F. Sadlo, "Visualization methods for vortex rings and vortex breakdown bubbles," in *EuroVis Proceedings,* 2007, pp. 211-218.

[34] M. Jankun-Kelly, J. Ming, D. Thompson and R. Machiraju, "Vortex Visualization for Practical Engineering Applications," *IEEE Transactions on Visualization and Computer Graphics,* vol. 12, pp. 957-964, September, 2006.

[35] F. Sadlo, R. Peikert and M. Sick, "Visualization Tools for Vorticity Transport Analysis in Incompressible Flow," *IEEE Transactions on Visualization and Computer Graphics,* vol. 12, pp. 949-956, 2006.

[36] R. S. Laramee, C. Garth, H. Doleisch, J. Schneider, H. Hauser and H. Hagen, "Visual analysis and exploration of fluid flow in a cooling jacket," in *IEEE Visualization '05,* 2005, pp. 623-630.

[37] D. Bauer, R. Peikert, M. Sato and M. Sick, "A case study in selective visualization of unsteady 3D flow," in *VIS '02: Proceedings of the Conference on Visualization '02,* Boston, Massachusetts, 2002, pp. 525-528.

[38] Q. You, S. Fang and L. Zhu, "Visualizing vortex shedding of an elastic plate interacting with a 3D viscous flow," in *CIT '09: Proceedings of the 2009 Ninth IEEE International Conference on Computer and Information Technology,* 2009, pp. 312-317.

[39] I. V. Pivkin, E. Hueso, R. Weinstein, D. H. Laidlaw, S. Swartz and G. E. Karniadakis, "Simulation and visualization of air flow around bat wings during flight," in *Proceedings of the International Conference on Computational Science*, 2002.

[40] R. Mittal, H. Dong, M. Bozkurttas, F. M. Najjar, A. Vargas and A. von Loebbecke, "A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries," *Journal of Computational Physics,* vol. 227, pp. 4825-4852, May, 2008.

[41] H. Dong, R. Mittal and F. M. Najjar, "Wake topology and hydrodynamic performance of low aspect-ratio flapping foils," *Journal of Fluid Mechanics,* vol. 566, pp. 309-343, November, 2006.

[42] R. Haimes and D. Kenwright, "On the velocity gradient tensor and fluid feature extraction," in *AIAA 14th Computational Fluid Dynamics Conference,* 1999.

[43] V. Verma, D. Kao and A. Pang, "A flow-guided streamline seeding strategy," in *IEEE Visualization '00*, Salt Lake City, Utah, United States, 2000, pp. 163-170.

[44] X. Ye, D. Kao and A. Pang, "Strategy for seeding 3D streamlines," in *IEEE Visualization '05*, 2005, pp. 471-478.

[45] C. Yuan, J. D. Cohen and J. H. Krolik, "Similarity-Guided Streamline Placement with Error Evaluation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 1448-1455, 2007.

[46] L. Li and H. Shen, "Image Based Streamline Generation and Rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 630-640, May-June, 2007.

[47] S. Marchesin, C. Chen, C. Ho and K. Ma, "View-Dependent Streamlines for 3D Vector Fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, pp. 1578-1586, November, 2010.