FlowVisual: A Visualization App for Teaching and Understanding 3D Flow Field Concepts

Man Wang; Michigan Technological University; Houghton, MI
Jun Tao; University of Notre Dame; Notre Dame, IN
Jun Ma; Michigan Technological University; Houghton, MI
Yang Shen, Chaoli Wang; University of Notre Dame; Notre Dame, IN

Abstract

The study of fluid behaviors has been a challenging topic. Flow visualization enables us to visually acquire qualitative and quantitative flow information. There exist various software tools performing different flow visualization tasks. However, we lack tools that help students learn important flow field concepts. In this paper, we present a visualization app, named FlowVisual which runs on iOS devices, to illustrate basic flow field concepts in 3D. In order to meet a comprehensive learning goal for students, we integrate a number of techniques into FlowVisual design, including field-line tracing, field-line comparison, critical point detection and classification, template-based seeding, and surface visualization. We evaluate and demonstrate the effectiveness of FlowVisual by conducting a formal user study including an introduction and training session, an auto-grading test, and a post-questionnaire survey.

Introduction

Fluid mechanics and computational fluid dynamics (CFD) are among the core courses in many engineering majors such as mechanical engineering, aerospace engineering, biomedical engineering, chemical engineering, and civil engineering. In these courses, it is important for students to acquire the knowledge of fundamental flow field concepts. Many of those concepts are not straightforward to learn. For instance, it is not easy for beginning-level students to fully understand the differences between various kinds of field-lines and critical points. Commonly, these materials are taught by instructors through explaining concepts and definitions, drawing diagrams and illustrations, and occasionally, playing custom-made animations or video clips. Using intuitive and real flow examples proves to be an excellent way of learning. However, most examples available today are only designed for lecture or demonstration but not for student interaction or self-learning. Developing a pedagogical visualization tool holds the potential to help students better learn these essential flow field concepts through interactive exploration.

In this paper, we present FlowVisual, an educational app running on iOS devices, to illustrate basic flow field concepts in 3D. This app is an extension of the desktop version of FlowVisual for 2D flow fields [14]. The desktop version has been used in classroom teaching of CFD course for multiple times and has received positive feedback from students. From our user study, we found that the app helped students with no previous 2D flow field knowledge understand concepts to the similar degree of students who had studied those concepts before. This new mobile FlowVisual is developed to illustrate the concepts in 3D space as cases in 3D are more common yet more challenging to understand in practice. Besides different kinds of field-lines, we also implemented stream surfaces in this app to enrich the perception of the flow field characteristics in a more continuous fashion. Our key deliverable is an app for classroom demonstration and for self-study by students and professionals. Its implementation on iPad makes it highly portable and accessible by anyone who is interested in learning and exploring key flow field concepts. The app has been used in a classroom environment and its effectiveness was evaluated through a formal user study involving students from mechanical engineering, electrical engineering, and computer science. The mobile FlowVisual is freely accessible in the App Store. The tutorial and evaluation materials are also available online so that instructors and students who are interested in our work can make use of them.

Related Work

Flow visualization plays a vital role in many scientific, engineering, and medical disciplines, offering users a graphical representation of their vector data for visual understanding, interpretation, and decision-making. For over two decades, flow visualization has been a central topic in scientific visualization, and a variety of techniques including glyph-based [10], texture-based [6], integration-based [8], topology-based [7], partition-based [12], and illustration-based [1] visualizations have been presented. Our tool focuses on integration-based flow visualization as it is most widely used in practice. For integration-based flow visualization, particles or seeds are placed in a vector field and advected over time. The traces or field-lines that the particles follow, e.g., streamlines for steady flow and pathlines for unsteady flow, depict the underlying vector data.

Teaching the core concepts of fluid dynamics has not experienced significant changes over the years. A few published works discussed some recent advances. Hertzberg and Sweetman [5] designed a flow visualization course focusing on studio/laboratory experiences for mixed teams of students. The course content includes fluid flow physics, history of photography with respect to the relationship with science and art, as well as flow visualization and photography techniques. Their course proved to be very successful in attracting both graduate and undergraduate students, engineering women in particular. Settles et al. [13] argued that fluid mechanics is fundamentally visual, and visual topics can be taught by modern multimedia methods. They described a new series of 10-15 minutes narrated videos that use flow visualization to illustrate basic fluid mechanics concepts. Rossmann and Skvirsky [11] developed a sophomore-level seminar that exposes

students to flow visualization techniques and the science of fluid mechanics, and to the photographic methods needed to create effective images. The fundamentals of fluid flow and photography were taught and practiced in a studio setting. As an interactive visualization app for learning flow field concepts, our FlowVisual builds on the solid education tool of iPad and provides an alternative to the above methods.

Terms

We give a brief introduction to some important concepts of flow fields. These concepts are incorporated into our FlowVisual design.

Flow Field A flow field (or vector field) is an assignment of a velocity vector to each point in the domain to represent the movement of the flow. Essentially, it is a mapping

$$F(\mathbf{p};t) = \mathbf{v} \tag{1}$$

that assigns a vector $\mathbf{v}$ to each point $\mathbf{p}$ at time t. Mathematically, a flow field could be expressed as a differential equation

$$\frac{d\mathbf{p}}{dt} = \mathbf{v}(\mathbf{p};t). \tag{2}$$

Steady and Unsteady Flow When all the time derivatives of a flow field vanish, the flow is considered to be a steady flow. In other words, steady flow refers to the condition where the fluid properties at each point in the system do not change over time. When time does affect the behavior of the flow, we consider the flow as an unsteady flow.

Streamline A streamline is the trajectory that a massless particle follows if released in a steady flow field. It is also known as the curve that is everywhere tangent to the vectors it passes through. Mathematically, a streamline is the solution from $\mathbf{p}_c = ((x_c, y_c, z_c); t_c)$ constrained in an instantaneous vector field of $\mathbf{v}(\mathbf{p}_c;t_c)$ at time t_c, and it can be represented by

$$\mathbf{p}(b) = \mathbf{p}_c + \int_0^b \mathbf{v}(\mathbf{p}(\sigma);t_c)d\sigma. \tag{3}$$

Pathline A pathline is the trajectory that an individual particle follows in an unsteady flow field. Given a flow field $\frac{d\mathbf{p}}{dt} = \mathbf{v}(\mathbf{p};t)$, the solution with initial state $\mathbf{p}_0 = ((x_0, y_0, z_0); t_0)$ is

$$\mathbf{p}(t = b) = \mathbf{p}_0 + \int_0^b \mathbf{v}(\mathbf{p}(\sigma);t_0 + \sigma)d\sigma, \tag{4}$$

which is referred to as a pathline starting at position $\mathbf{p}_0$.

Streakline A streakline is the locus of points of all the fluid particles that have passed continuously through a particular spatial point in the past. Given a set of pathlines traced from the same position at different time steps, connecting all points at the same time step forms a streakline.

Timeline A timeline is a line formed by a set of fluid particles that were marked at a previous instant in time, creating a curve that is displaced over time as the particles evolve. Given a set of pathlines traced from the same time step at different positions, connecting all the points at the same time step forms a timeline.

Stream Surface A stream surface is a continuous surface that is everywhere tangent to the vector it passes through, which can be obtained from streamlines traced from a densely seeded curve.

Critical Point A point $\mathbf{p}$ is called a critical point of $\mathbf{v}(\mathbf{p};t_c)$ if $\mathbf{v}(\mathbf{p};t_c) = 0$. Critical points are crucial because they are enclosed by their compact neighborhood with distinct patterns determined by their types.

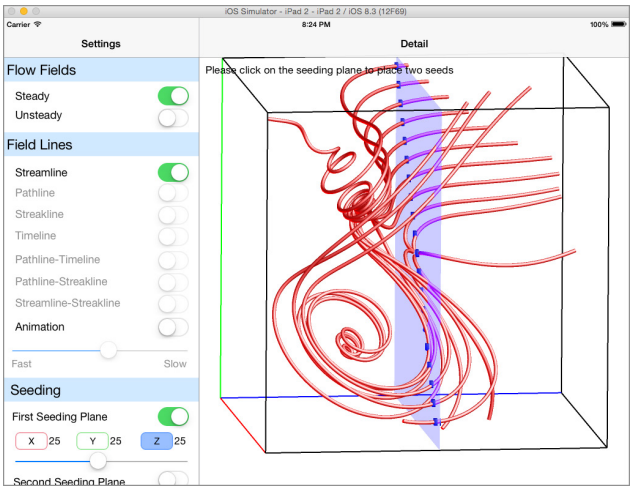
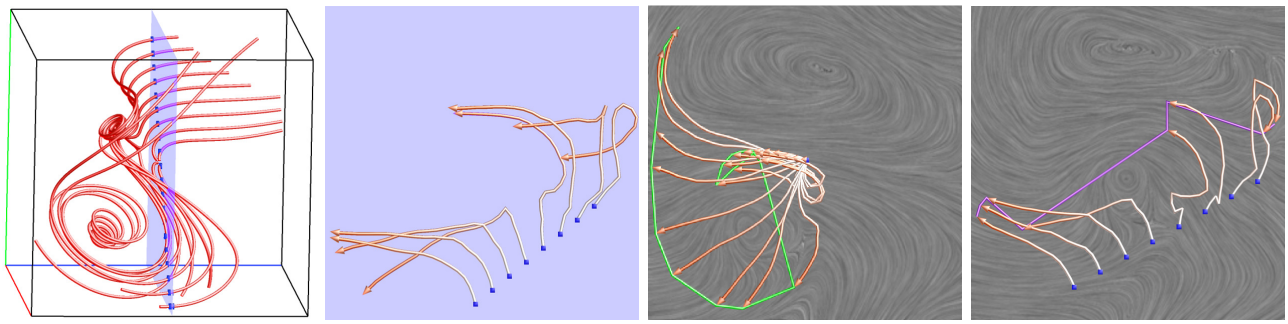


Figure 1. *The user interface of the FlowVisual app.*

Overview

Figure 1 shows the user interface of the FlowVisual app. There are two major parts: a drawing canvas and a function panel. The drawing canvas is where users place seeds and where flow field concepts are visualized. The panel has three sections: field-line section, stream surface section, and critical point section. The field-line section supports visualization of different field-lines to help students understand the definitions and their similarities and differences. It includes the visualization of streamline, pathline, streakline, and timeline, as well as the comparisons of pathline and streakline, pathline and timeline, and streamline and streakline. We support two types of seeding: point seeding and rake seeding. In addition, line integral convolution (LIC) textures can be displayed, which provides an overview of the underlying flow to guide seed placement. The stream surface section supports multiple surface overview and single surface inspection with streamlines and streamline animation. The multiple surface overview provides an overall impression of the flow field by displaying multiple stream surfaces at the same time. The single surface inspection allows one surface to be examined along with streamlines and streamline animation, without occlusion from other surfaces. The critical point section supports the detection and classification of critical points and template-based seeding. We carefully place seeds around each critical point to reveal the flow pattern in its vicinity.

We use a hurricane simulation data set and a synthesized five critical points data set in this app. The hurricane data set is made available through IEEE Visualization 2004 Contest. The data set is downsampled and used to illustrate flow field concepts for un-



(a) (b) (c) (d)

Figure 2. *Field-lines. (a) streamlines, (b) pathlines, (c) pathline-streakline with LIC, and (d) pathline-timeline with LIC.*

steady flow fields. The five critical points data set is made available by Dr. Alex Pang through his IEEE Visualization 2005 paper [16]. This steady flow field data set is mainly used for the demonstration of critical points and stream surfaces.

Functions and Implementation

Field-lines

Seeding To trace field-lines, users may specify any point in the domain as the seed. To ease the placement of seeds in the 3D space, we use seeding planes to fix the coordinate of one dimension and allow users to place seeds on the seeding planes. Users can use up to two seeding planes simultaneously. Either of the two planes can be switched between xy, yz or xz planes with adjustable z, x or y coordinates, respectively. Our app supports both point seeding and rake seeding. Point seeding allows only one seed at a time. Rake seeding, on the other hand, allows multiple seeds to be placed. Users are asked to click two end points. Based on the number of seeds (between 2 and 20) specified on the user interface, seeds will be placed evenly in between the chosen two end points.

Line Drawing To trace the trajectory of a particle within a flow field, we need to solve for positions that the particle passes through using the differential equation representing the flow field. We employ the fourth-order Runge-Kutta (RK4) method which numerically integrates ordinary differential equations by using a trial step at the midpoint of an interval to cancel out lower-order error terms. We depict the field-lines in two forms: solid tubes (see Figure 1) and animated arrows (see Figure 4). The tube form is depicted by simply connecting the points that we trace along the field-line. It shows the entire field-line with different colors representing different types of field-lines. As an extension of the static tube, animated arrows show the formation of filed-lines. In addition to drawing the entire line as the background using a transparent gray color, the animation uses dashed lines with arrowheads to indicate flow directions.

Visualization and Comparison Our app includes the visualization of four different types of field-lines: streamline, pathline, streakline, and timeline. To distinguish different field-lines, we employ distinct colors: red for streamlines, orange for pathlines, green for streaklines, and purple for timelines. Streamlines are traced in one single time step (i.e., steady field), while all other field-lines are traced over multiple time steps (i.e., unsteady field). Figure 2 shows different field-lines depicted by our app.

To demonstrate the concepts of field-lines and their relationships, we also provide multiple field-lines comparisons with animation. Timeline and streakline are both defined upon pathlines. Therefore, including pathline-timeline comparison and pathline-streakline comparison by showing the formation of timeline and streakline step by step would help users better understand the relationships between these flow lines.

Additionally, we use the LIC texture as the background to provide an overview of the flow within a plane in the flow field. The algorithm for generating LIC texture was introduced by Cabral and Leedom [2]. It works by adding a random static pattern of black-and-white paint sources to visualize the flow field. As the flow passes by the sources each fluid particle picks up some of the source intensity. The result is a striped texture where points along the same streamline tends to have similar intensities. If the LIC texture is turned on when tracing pathline, streakline, or timeline, the texture will be updated synchronously over time showing the underlying unsteady flow field. Users can adjust the animation speed as desired. Figures 2 (c) and (d) show pathline-streakline and pathline-timeline comparisons at selected time steps with LIC textures.

Stream Surfaces

A stream surface is a continuous surface that is everywhere tangent to the vector it passes in a steady flow field. It can be obtained by connecting the set of streamlines traced through every sample point on a seeding curve. In contrast to having numerous discrete streamlines in an area, a surface presents the flow pattern in a more coherent manner.

Seeding and Surface Construction Selecting the seeding curve for a surface is crucial for surface generation. It influences the resulting surface in two aspects: the effectiveness of surface in characterizing flow features, and the smoothness of the surface.

We choose the seeding curves whose corresponding stream surfaces are able to capture the pattern of critical points. The types and locations of critical points reveal important patterns of a flow field, which are difficult to predict if no surfaces pass through those regions. Once the starting seed is placed, the following seeds on the curve are generated along the binormal vector of the previous seed so that the resulting stream surface can demonstrate

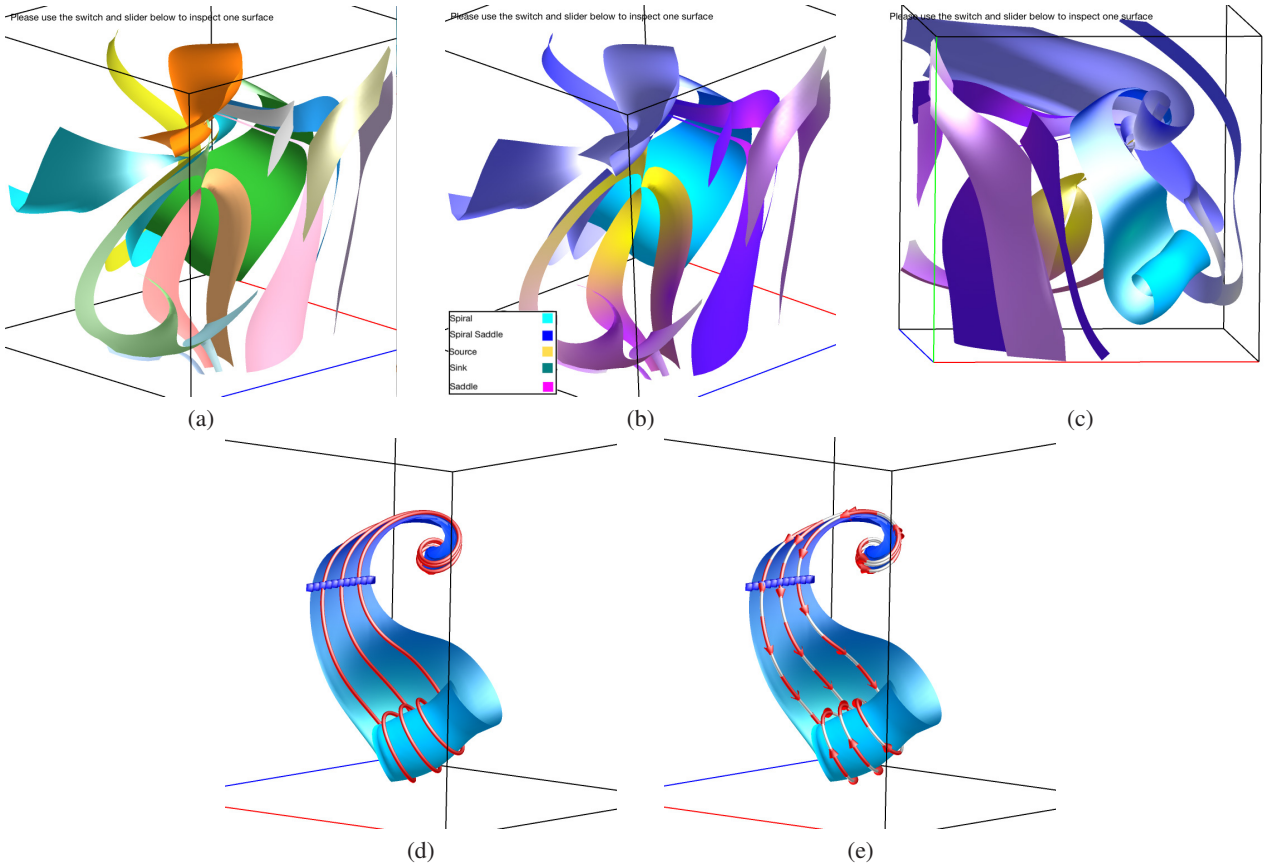


Figure 3. *Stream Surfaces. (a) multiple surface overview with unique color for each surface, (b)-(c) multiple surface overview with coloring based on the types of related critical points, (d) single surface with streamlines, and (e) single surface with streamline animation.*

the flow direction. For a point **p** on a streamline with velocity vector **v** and normal vector **n**, its binormal vector is the vector at **p** that is orthogonal to the plane containing **v** and **n**.

To generate smooth surfaces, we use a threshold value σ as the largest distance between two consecutive sample points to ensure the seeding curve is densely sampled. Then, we employ the easy integral surface algorithm [9] for surface construction where the maximum distance between two consecutive sample points on the propagation front is within σ. The front propagation is performed by tracing streamlines one step at a time and connecting the neighboring points into quads. Special cases such as divergence and convergence are taken care of to fill the gaps and avoid oversampling.

The surfaces are manually selected for the given five critical points data set. We first randomly generate 3000 lines that follow the binormal direction, and then manually specify segments of these lines as the seeding curves. To capture the flow features, we select those surfaces that pass through the critical regions. On the other hand, to avoid the surface being overly complicated, we do not select those surfaces that diverge and end at more than two critical points.

Surface Coloration To show the correspondence between stream surfaces and critical points, we color stream surfaces that relate to the same critical points with similar colors (see Figure 3 (b) and (c)). As mentioned before, a single surface passes no more than two critical points. Given zero velocity at critical points, the tracing of a surface either terminates at a critical point or on the boundaries of the flow field. This indicates that each of our surfaces connects two critical points or is constructed between a critical point and the volume's boundary. In the former case, the color of a surface vertex is linearly interpolated between the colors of the two critical points. In the later case, the color will gradually fade out as the surface moves far away from the critical point.

Surface Drawing We precompute a total of fourteen surfaces from the five critical points data set and store them as files to reduce runtime workload and ensure prompt response during interaction. There are two options to examine the surfaces: multiple surface overview, and single surface inspection with streamlines and streamline animation. The first option displays all 14 surfaces to present an overview of the flow field. The second option enables closer inspection of a particular surface with streamlines or streamline animation.

Streamline Drawing In the single surface inspection mode, we provide two options: streamlines and streamline animation, to help detailed inspection. An example is shown in Figure 3 (d) and (e). The displayed surface will have three streamlines evenly distributed on the surface to show the exact pattern of the flow.

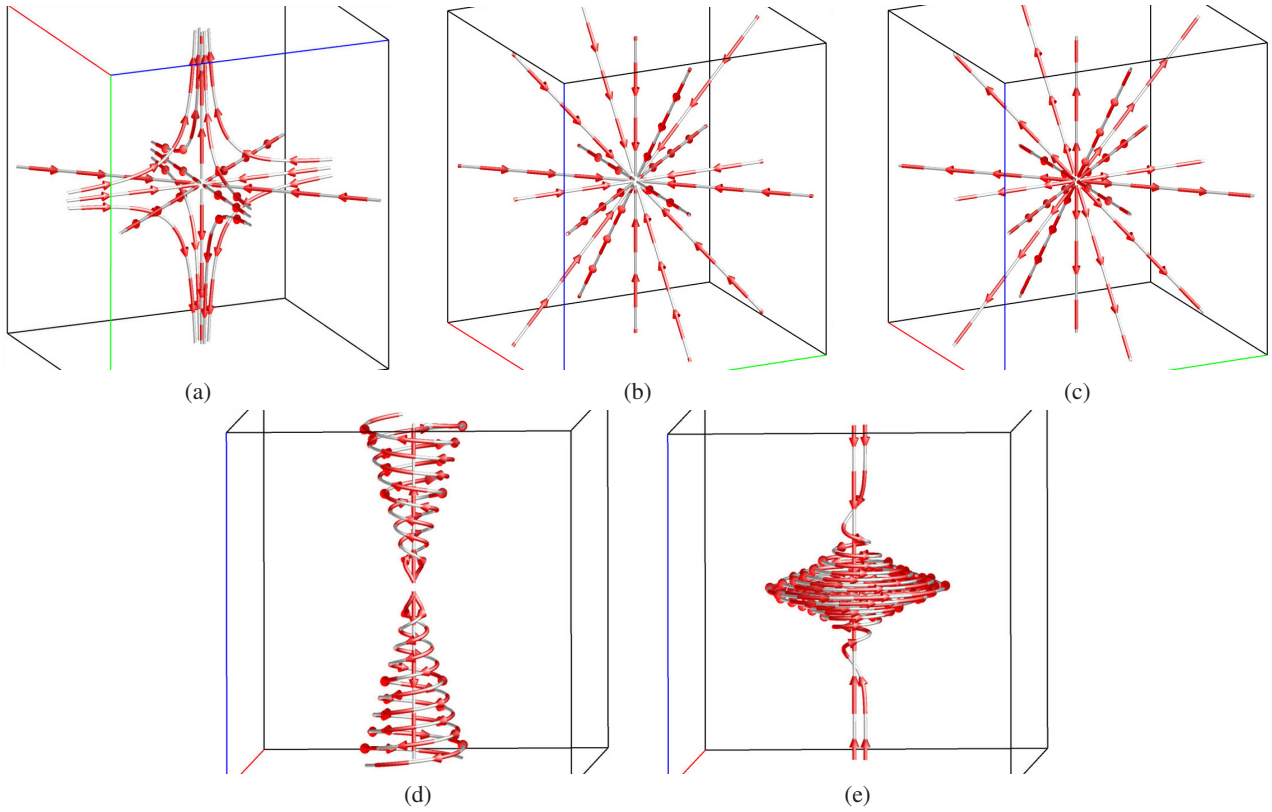


Figure 4. *Critical points and their seeding templates. (a) saddle, (b) sink, (c) source, (d) spiral, and (e) spiral saddle.*

Streamline animation is also available for showing the speed and direction of the flow.

Critical Points

Extracting features from flow fields has been a topic of active research for decades. A great deal of work has been done to tackle this problem. Given a flow field, we achieve the following two goals in this work: figuring out locations and types of critical points for a given flow field, and designing seeding templates that effectively capture streamline patterns around different types of critical points.

Detection A critical point is a position in a flow field domain where the velocity vanishes. For discrete vector data, we detect critical points through sign checking and vector interpolation. Specifically, for each voxel, we check whether there is at least one change of sign of the vectors at its corners. If the x, y and z vector components all have a sign change, it means that a critical point may exist within the voxel and a further step is taken to obtain the precise location of the critical point.

For a 3D flow field, the detection of critical points is achieved by utilizing the Greene's bisection method [3]. This method divides the flow field into equally-sized cubes and computes their Poincaré index to check the existence of any critical point within the cube. If a cube has a non-zero Poincaré index, the cube will be bisected into subcubes iteratively to find the precise locations of the critical points.

Classification Critical points are classified according to the flow patterns in their neighborhood. Mathematically, the type of critical point is determined by the real and imaginary parts of the eigenvalues of the Jacobian matrix in the neighborhood of the critical point. Since imaginary parts demonstrate the circulating flow pattern while real parts represent the repelling or attracting behavior of the flow, an analysis on both parts would determine the types of critical points [4, 15].

A first-order critical point $\mathbf{p}_0$ where $\mathbf{v}(\mathbf{p}_0 : t) = 0$ can be classified based on its eigenvalues of the Jacobian matrix $\mathbf{J}_\mathbf{v}(\mathbf{p}_0)$ when $det(\mathbf{J}_\mathbf{v}(\mathbf{p}_0)) \neq 0$.

Consider a 3D flow field

$$\mathbf{v}(x,y,z) = \begin{pmatrix} u(x,y,z) \\ v(x,y,z) \\ w(x,y,z) \end{pmatrix},$$

we have its Jacobian matrix as follows

$$\mathbf{J}_\mathbf{v}(x,y,z) = \begin{pmatrix} \frac{\partial u(x,y,z)}{\partial x} & \frac{\partial u(x,y,z)}{\partial y} & \frac{\partial u(x,y,z)}{\partial z} \\ \frac{\partial v(x,y,z)}{\partial x} & \frac{\partial v(x,y,z)}{\partial y} & \frac{\partial v(x,y,z)}{\partial z} \\ \frac{\partial w(x,y,z)}{\partial x} & \frac{\partial w(x,y,z)}{\partial y} & \frac{\partial w(x,y,z)}{\partial z} \end{pmatrix}.$$

Let λ_1, λ_2, λ_3 be the eigenvalues of $\mathbf{J}_\mathbf{v}(\mathbf{p}_0)$. R_1, R_2, R_3 are their real parts, and I_1, I_2, I_3 are their imaginary parts. We order λ_1, λ_2,

λ_3 according to the values of their real parts so that $R_1 \leq R_2 \leq R_3$. Based on the sign of their real parts and the presence of imaginary parts, critical points can be grouped into the following types:

Repelling node:	$R_{1,2,3} > 0$	$I_{1,2,3} = 0$
Attracting node:	$R_{1,2,3} < 0$	$I_{1,2,3} = 0$
Repelling focus:	$R_{1,2,3} > 0$	$I_1 = 0, I_{2,3} \neq 0$
Attracting focus:	$R_{1,2,3} < 0$	$I_1 = 0, I_{2,3} \neq 0$
Repelling node saddle:	$R_1 < 0 < R_2 \leq R_3$	$I_{1,2,3} = 0$
Attracting node saddle:	$R_1 \leq R_2 < 0 < R_3$	$I_{1,2,3} = 0$
Repelling focus saddle:	$R_1 < 0 < R_2 \leq R_3$	$I_1 = 0, I_{2,3} \neq 0$
Attracting focus saddle:	$R_1 < 0 < R_2 \leq R_3$	$I_1 = 0, I_{2,3} \neq 0$
Center:	$R_{1,2,3} = 0$	$I_1 = 0, I_{2,3} \neq 0$

For simplicity, we merge all types of critical points into five types in our app as shown in Figure 4. Below is our merging strategy:

Source:	Repelling node
Sink:	Attracting node
Spiral:	Repelling focus
	Attracting focus
Spiral saddle:	Repelling node saddle
	Attracting node saddle
	Repelling focus saddle
	Attracting focus saddle

This strategy preserves the distinction of sink and source. It also categorizes the types whose patterns present a combination of spiral and saddle shapes. In this way, we avoid using too many different colors to differentiate different types of critical points on the screen. As a result, users will not have to constantly refer back and forth to check the color and the type of a critical point.

Template Design Since streamline patterns around distinct types of critical points are quite different from one another, streamline placement becomes important in order to effectively reveal the characteristics of critical points. We adopt a similar strategy proposed by Ye et al. [16] that applies a different seeding template for each type of critical point. For a saddle, the template has the seeds distributed along the bisector of the hyperbola. But for an attracting/repelling focus or a center, the pattern forms a circular or spiral shape. Therefore, seeding on concentric circles with increasing radii is an appropriate strategy. As for an attracting/repelling node, the flow directions are either toward or away from the critical point. The corresponding strategy is to place seeds evenly on a circle centered at the critical point.

Visualization Our app allows users to turn on/off a type of critical point. Critical points are drawn in the color of their types as indicated on the interface. By clicking on a critical point, its template seeds are placed to trace the corresponding streamlines so that the flow field around the critical point could be perceived. By applying a different template for each type of critical point, we avoid the overwhelming occlusion brought by drawing all streamlines around critical points while maintaining interactive performance. In addition, users may edit the templates through changing the size of the template, the number of layers along the z direction, and the distance between the layers. Figure 5 shows examples of visualizing critical points and their corresponding streamlines.

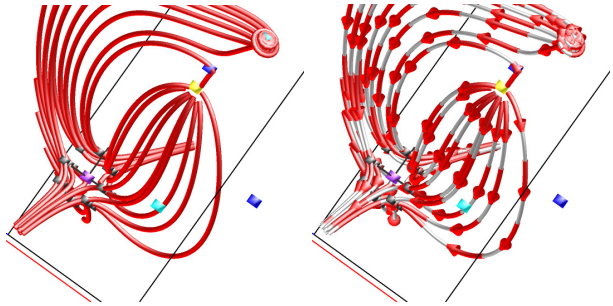


Figure 5. *Streamlines around critical points via template-based seeding. (a) static streamlines and (b) dynamic streamlines with animated arrows.*

Auto-grading

We design an auto-grading component to evaluate a student's understanding of flow field concepts. We provide four kinds of questions:

- The first kind is text-based questions asking about the core concepts involved.
- The second kind lists some visualization results generated from our app and asks which one corresponds to which type of field-line. Here the visualization results use a different field-line coloring scheme in order to eliminate the possible memorization of field-line colors encoded by our app.
- The third kind asks students to classify the type for a given critical point or locate a critical point with a given type. Students need to trace streamlines over the flow field in order to answer these questions correctly.
- The fourth kind asks students to locate a certain type of critical point or tell the types of critical points given stream surfaces with and without streamlines and streamline animation.

The auto-grading function was used for user study in our work. The answers and the grade of each student taking the test are stored in a file, which is sent to the instructor via email. This component could also be used in class assignment or quiz to test students' understanding and help instructors' in their course design.

Implementation

The mobile FlowVisual interface was developed using Xcode provided by Apple in Objective C. It includes the panel on the left for various widgets and the canvas on the right for visualization. The visualization was implemented using OpenGL ES 2.0. This version of OpenGL ES has a programmable pipeline, which requires the use of shaders and vertex buffer object (VBO). The flags of attributes such as lighting, texture, the matrices for viewport setting and the values of texture, as well as the coordinate, color, normal vector, and index of each point to draw must

be attached to some variables and VBO buffers. The vertex and fragment shaders will then be compiled. To display any item on the canvas, the coordinates, colors, normal vectors, and indices of the points are computed and loaded into the related VBO buffer.

Due to the limited memory and computation power available on iPad, the storage of inherently large flow field data and the intense and frequent computation of LIC textures on the fly can be difficult to achieve. We therefore adopted several ways to overcome these challenges. To reduce the size of the flow field data, we down-sampled the data spatially and reduced the number of time steps loaded for an unsteady flow field. To avoid LIC texture computation on the fly, we precomputed LIC textures and saved them as JPG files with affordable extra memory cost. The seeding curve and surface data were also precomputed and saved into files since generating stream surfaces at run time can be costly which would negatively impact the performance.

Evaluation

We conducted two user studies to evaluate the effectiveness of the mobile FlowVisual: one for the sections of field-line and critical point, and the other for stream surface section. The main reason for conducting two separate user studies is because surface visualization was incorporated in a later stage as an update of the app. Both studies consist of two parts, an auto-grading test using our app and a paper-based survey. The auto-grading test asks students to interact with the app and answer related questions. The students were informed that the main factor for evaluation was how the app can improve their understanding of flow field concepts. The studies were conducted in our lab. Before a test started, the students were first briefed with an introduction to the flow field concepts and our app, and were given time to explore the app. When they felt comfortable with the concepts and the app, they could perform the auto-grading test and fill in the survey. During the test, the students were only allowed to ask questions regarding the use of the app. Test questions were presented one by one. Once a question was answered, the student could not go back for review or correction.

The first survey on field-line and critical point includes 21 rating questions and six open questions. The second survey on stream surface has eight rating questions and three open questions. The rating questions ask the participants to evaluate the presentation of field-lines and critical points, stream surfaces, and the design of the user interface (UI). Each function (e.g., streamline point seeding, streamline rake seeding, streakline animation, etc.) has a question that asks the students to rate how much they agree that a particular function helps them understand the concept. The available choices are 5: strongly agree, 4: agree, 3: neutral, 2: disagree, and 1: strongly disagree. Based on the nature of the content, the questions are divided into the groups. The question groups in the first survey are “flow field”, “streamline”, “pathline”, “streakline”, “timeline”, “animation”, “critical point”, “UI color”, and “UI size”, where the “UI size” category has questions on the size of seeds and critical points, and the width of lines. Each group has two to five questions, and one question may appear in multiple groups. For example, a question asking the students to rate the streamline animation capability appears in both the streamline group and the animation group. The questions in the second survey have groups of “flow field”, “stream surface alone is better than streamlines with rake seeding”, “stream surface with streamlines is better than stream surface alone”, “stream surface with streamline animation is better than stream surface alone”, “stream surface with streamline animation is better than stream surface with streamlines”, and “UI color”.

All students recruited in both user studies are from Michigan Technological University. Each student was paid $10 in return. There were 21 students participated in the first study. Ten of them majored in electrical engineering and computer science (EECS) with one undergraduate student, and eleven were from mechanical engineering (ME) department with two undergraduate students. Some ME students, especially graduate students, have taken the CFD course, where the students learned the basic concepts for kinematics and dynamics of fluid flows. The flow field concepts such as streamline and streakline were introduced at the beginning, followed by the introduction of the Navier-Stokes equations. The CFD course focused on the methods to analyze engineering problems, such as control volume analysis and non-dimensional analysis. For the second survey, there were 11 participants from varied majors. These students can be categorized into groups with CFD background (five students) and without CFD background (six students). One participant from the group without CFD background was identified as an outlier based on the auto-grading test results. The test results have the first quartile Q_1 of 3, median of 3, the third quartile Q_3 of 4 and thus its interquartile range (IQR) evaluates to the range between the first and third quartile, which is 1. Any data point that lies outside the range of [Q_1-1.5×IQR, Q_3+1.5×IQR], i.e., [1.5, 5.5] in our case, is considered as an outlier. Therefore, this participant with score of 1 was treated as an outlier and the feedback was not considered in the following analysis.

General Findings

The top table in Table 1 shows the mean and standard deviation of each question group from the first study. In general, student reactions to the app was positive with mean values larger than 4.40, which is a positive feedback to all the elements of the app.

The streakline had the largest standard deviation (0.82) perhaps for the reason of the inherent difficulty of its definition. We investigated the differences of student groups. The bottom table in Table 1 summarizes the ratings from EECS, ME, UG (undergraduate), and Grad (graduate) students. Students from MEG performed best amongst all groups. This may be due to their previous study in fluid dynamics and more in-depth research experience compared with undergrads from the same department. We also find that students from ME performed slightly better than students from EECS and the mean values of group EECS and MEUG were quite close.

Table 2 shows the mean and standard deviation of each question group from the second study. Groups except the “stream surface alone is better than streamlines with rake seeding” received average ratings no less than 4.40. Group “stream surface alone is better than streamlines with rake seeding” received the lowest score of 3.70 with the largest standard deviation of 0.94. This is, however, not surprising since a stream surface alone is the combined depiction of many streamlines. Overall, we received a positive feedback on the stream surface section and the stream surface with streamline animation is the most preferred visual representation.

	flow field	stream line	path line	streak line	timeline line	animation	critical point	UI color	UI size	EECS	ME	UG	Grad
mean	4.42	4.44	4.52	4.49	4.73	4.55	4.44	4.45	4.69	4.53	4.49	4.48	4.52
std dev	0.69	0.71	0.72	0.82	0.45	0.70	0.61	0.63	0.52	0.64	0.63	0.50	0.65

	EECS	ME	UG	Grad	MEUG	MEG	all
mean	72.78	76.26	72.22	74.85	73.61	79.08	74.60
std dev	17.85	21.62	17.07	7.86	5.56	14.51	15.86

Top: means and standard deviations of field-line and critical point question groups and student groups. Bottom: means and standard deviations of field-line and critical point auto-grading test results.

	flow field	surface alone better than lines with rake seeding	surface + lines better than surface alone	surface + line animation better than surface	surface + line animation better than surface + lines	UI color	w/o CFD background	w CFD background
mean	4.20	3.70	4.40	4.70	4.40	4.55	4.20	4.43
std dev	0.61	0.94	0.84	0.67	0.69	0.48	0.75	0.71

	w/o CFD background	w CFD background	all
mean	80.00	85.00	82.50
std dev	20.92	13.69	16.87

Top: means and standard deviations of surface question groups and student groups. Bottom: means and standard deviations of surface auto-grading test results.

Study of Group Differences with MANOVA

For each group of questions, we investigated the significance of mean difference between EECS and ME, between UG and Grad, and between MEUG and MEG for the first survey and between groups of with and without CFD background for the second survey by applying MANOVA using the Wilks' λ test. We used the significance level of 0.05, meaning that the null hypothesis is rejected if the p-value is smaller than 0.05. From our analysis, all p-values from group pairs are greater than the significance level. Therefore, the null hypothesis can not be rejected and no significant difference was found between the ratings of the two groups. This shows that student ratings were very similar in spite of their discipline and level of education differences.

Student Comments

The open questions were designed to allow the students to provide detailed comments about the reasoning behind their ratings and suggestions for future development of our app. The overall feedback was positive. Some students stated that "*It helped me understand different types of field-lines and the animation of two types of field-lines impressed me*", "*The types of line/point were distinguished clearly by colors*", and "*Animation effectively showed the motion of the flow*". They also mentioned that "*Different types of critical points are very easy to tell using this tool*", "*Clicking on a critical point and displaying its pattern is very helpful*", and "*Stream surface helps a lot to catch the patterns and identify the features of the flow field*".

Most students were satisfied with the user interface. We also received some valuable suggestions. These included relocating the hint to the top of the screen, using a different font size or color, and coloring the surfaces according to the types of related critical points. All these suggestions have been incorporated into the latest version of the app. In the version used in the user study, each surface uses a unique color to distinguish itself from other surfaces. This provides participants with no hints to answer the auto-grading questions as they need to entirely rely on the shapes of surfaces to determine the type of a critical point. Figure 3 (a) shows the original coloring scheme used in the user study, and Figure 3 (b) and (c) display the surfaces colored according to the related critical points as suggested by the participants.

Auto-grading Test

The auto-grading test assessed student learning and automatically checked the correctness of their work. The test score of each student was rescaled to [0, 100]. The bottom table in Tables 1 and 2 show the mean values and standard deviations of student groups. Generally, students with CFD background/ME students performed better than students without CFD background/EECS students. However, the scores were reasonably close to an overall average score. In the first study, students without CFD background have an average score of 72.78 with the overall average of 74.60 from all students. In the second study, students without CFD background have an average of 80.00 with the overall average of 82.50. The questions in the first study were categorized into twelve groups: text-based, image/animation-based, flow field basics, streamline, streakline, pathline, timeline, straightforward concepts, profound understanding, classification of saddle points, classification of other critical points, and location of critical points. MANOVA tests did not report significant differences for question groups between EECS and ME, between UG and Grad, or between MEUG and MEG. In addition, we tested the significance among these student groups for each individual question using ANOVA. The smallest p-value was 0.1411 at Q4. We also investigated the possible differences caused by major and education level using MANOVA on question groups. The p-value from MANOVA test is 0.8912. Therefore, no significant difference was found from MANOVA or ANOVA tests at the 0.05-level.

We conducted MANOVA and ANOVA tests on the auto-grading results from the second study to compare the two student groups with different levels of background on fluid dynamics. We

did not find any significant difference at the 0.05-level either.

In summary, FlowVisual received an overall positive feedback and the students without CFD background/EECS students performed as well as the students with CFD background/ME students after using our app, even though the former had less background or knowledge of the field.

User Study Summary

Our user study indicated that FlowVisual was effective in facilitating learning 3D flow field concepts and that the students were able to pick up the concepts efficiently. Therefore, regardless of the background of the students (with or without prior fluid dynamics knowledge), our visualization app turns out to be effective and helpful for classroom teaching and self-learning.

The rating scores indicated that reactions from the students were overall positive. During our user study, some students attended CFD course informed us that there was no interactive tool being used in the class when they were introduced to those concepts. As a result, most of them only had some vague impression. It is quite common having videos demonstration in classroom teaching. However, there are concerns from students that they would easily lose interest due to the lack of interaction. As the video content is fixed, it is difficult to know what would happen if they placed some seeds at certain spots in the flow field. These concerns no longer exist if an interactive visualization tool is used to assist teaching. In this regard, FlowVisual not only presents a tool for classroom teaching and demonstration, but also serves as an effective and efficient aid for self-study after class.

Conclusions

Our FlowVisual app provides various ways of visualizing and comparing flow field concepts. The single field-line visualization depicts individual concepts correctly and clearly. The comparisons between field-lines present the formations and the relationships of complicated field-lines. For example, FlowVisual helps to clarify the idea that timelines and streaklines are built based on pathlines. Stream surface provides another way to understand the flow field. It shows flow features coherently and clearly, enhancing the perception of flow patterns in conjunction with streamlines and streamline animation. Furthermore, users are able to learn and explore field-lines and stream surfaces interactively. Besides the basic concepts related to field-lines and stream surface, our app also provides critical point detection and classification. Seeding templates are applied to critical points to visualize the essential flow patterns in their neighborhood, presenting the unique influence of each type of critical point to the flow field. The feedback received from user evaluation is positive. It shows that our app is effective in facilitating learning 3D flow field concepts and the students are able to pick up the concepts efficiently.

FlowVisual is available at the App Store for free download. The app currently supports two data sets, one steady and one unsteady. The accompanying LIC images, seeding curves of stream surfaces, along with surface data are precomputed to reduce computation overhead on the mobile devices with constrained resource. In the future, we will provide the capability to support the importing of user-defined data sets.

Acknowledgments

This work was supported in part by the Dave House Family Foundation and the U.S. National Science Foundation through grants IIS-1017935, IIS-1456763, and IIS-1455886. We thank Dr. Alex Pang for his valuable advice on 3D critical point detection.

References

[1] A. Brambilla, R. Carnecky, R. Peikert, I. Viola, and H. Hauser. Illustrative flow visualization: State of the art, trends and challenges. In Eurographics State-of-the-Art Reports, pages 75—94, 2012.

[2] B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. In Proceedings of ACM SIGGRAPH Conference, pages 263—270, 1993.

[3] J.M. Greene. Locating three-dimensional roots by a bisection method. Computational Physics, 98(5):194—198, 1992.

[4] J. Helman and L. Hesselink. Representation and display of vector field topology in fluid flow data sets. IEEE Computer, 22(8):27—36, 1989.

[5] J. Hertzberg and A. Sweetman. A course in flow visualization: the art and physics of fluid flow. In Proceedings of American Society for Engineering Education Annual Conference, pages 2449—2459, 2004.

[6] R. S. Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F. H. Post, and D. Weiskopf. The state of the art in flow visualization: Dense and texture-based techniques. Computer Graphics Forum, 23(2):203—222, 2004.

[7] R. S. Laramee, H. Hauser, L. Zhao, and F. H. Post. Topology-based flow visualization, the state of the art. In H. Hauser, H. Hagen, and H. Theisel, editors, Topology-Based Methods in Visualization, chapter 1, pages 119. Springer Berlin Heidelberg, 2007.

[8] 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, 29(6):1807—1829, 2010.

[9] T. McLoughlin, R. S. Laramee, and E. Zhang. Easy integral surfaces: A fast, quad-based stream and path surface algorithm. In Proceedings of Computer Graphics International, pages 73—82, 2009.

[10] Z. Peng and R. S. Laramee. Higher dimensional vector field visualization: A survey. In Proceedings of Theory and Practice of Computer Graphics, pages 149—163, 2009.

[11] J. S. Rossmann and K. A. Skvirsky. You dont need a weatherman to know which way the wind blows: The art & science of flow visualization. In Proceedings of ASEE/IEEE Frontiers in Education Conference, 2010.

[12] T. Salzbrunn, H. Janicke, T. Wischgoll, and G. Scheuermann. The state of the art in flow visualization: Partition-based techniques. In Proceedings of Simulation and Visualization Conference, pages 75—92, 2008.

[13] G. Settles, G. Tremblay, J. Cimbala, L. Dodson, and J. Miller. Teaching fluid mechanics with flow visualization videos. In Proceedings of International Symposium on Flow Visualization, 2006.

[14] M. Wang, J. Tao, C. Wang, C.-K. Shene, and S. H. Kim. FlowVisual: Design and evaluation of a visualization tool for teaching 2D flow field concepts. In Proceedings of American Society for Engineering Education Annual Conference, 2013.

[15] T. Weinkauf, H. Theisel, H.-C. Hege, and H.-P. Seidel. Boundary switch connectors for topological visualization of complex 3D vector fields. In Proceedings of Joint Eurographics - IEEE TCVG Symposium on Visualization, pages 183—192, 2004.

[16] X. Ye, D. Kao, and A. Pang. Strategy for seeding 3D streamlines.

In Proceedings of IEEE Visualization Conference, pages 47—478, 2005.

Author Biography

Man Wang *is a Ph.D. student of computer science at Michigan Technological University. She received a Master degree in computer science from Michigan Technological University in 2013. Her current research interest is design and evaluation of pedagogical visualization tools for understanding fluid dynamics and computer security concepts. She received House Fellow award for the 2012-2013 academic year at Michigan Technological University. In the summers of 2012-2014, she was an instructor for the Summer Youth Programs.*

Jun Tao *is currently a postdoctoral researcher at University of Notre Dame. He received a Ph.D. degree in computer science from Michigan Technological University in 2015. Dr. Tao's major research interest is scientific visualization, especially on applying information theory, optimization techniques, and topological analysis to flow visualization and multivariate data exploration. He is also interested in graph-based visualization, image collection visualization, and software visualization. He received the Dean's Award for Outstanding Scholarship and the Finishing Fellowship at Michigan Technological University in 2015, and a Best Paper Award at IS&T/SPIE VDA 2013.*

Jun Ma *is a senior system engineer in Qualcomm, Inc. He received the Master and Ph.D. degrees in computer science from Michigan Technological University in 2009 and 2014, respectively. Dr. Ma's research interests include computer graphics and data visualization, particularly on flow visualization, big data visualization, and educational visualization. He received the Finishing Fellowship at Michigan Technological University in 2014, a Best Paper Award at IS&T/SPIE VDA 2015, and an Honorable Mention at IEEE PacificVis 2013.*

Yang Shen *is a Ph.D. student of computer science and engineering at University of Notre Dame. His research interests include visual analytics, data mining, and flow visualization. He received a BE degree in software engineering from Northwestern Polytechnical University, China, in 2012.*

Chaoli Wang *is an associate professor of computer science and engineering at University of Notre Dame. He received a Ph.D. degree in computer and information science from The Ohio State University in 2006. Prior to joining Notre Dame, he was a postdoctoral researcher at University of California, Davis and an assistant professor of computer science at Michigan Technological University. Dr. Wang's main research interest is scientific visualization, in particular on the topics of time-varying multivariate data visualization, flow visualization, and information-theoretic algorithms and graph-based techniques for big data analytics. He is a recipient of the NSF CAREER Award (2014), best paper awards at IS&T/SPIE VDA (2013, 2015), and an honorable mention at IEEE PacificVis (2013).*