Visualizing Flow Trajectories Using Locality-based Rendering and Warped Curve Plots

Chad Jones, *Student Member, IEEE*, and Kwan-Liu Ma, *Member, IEEE*

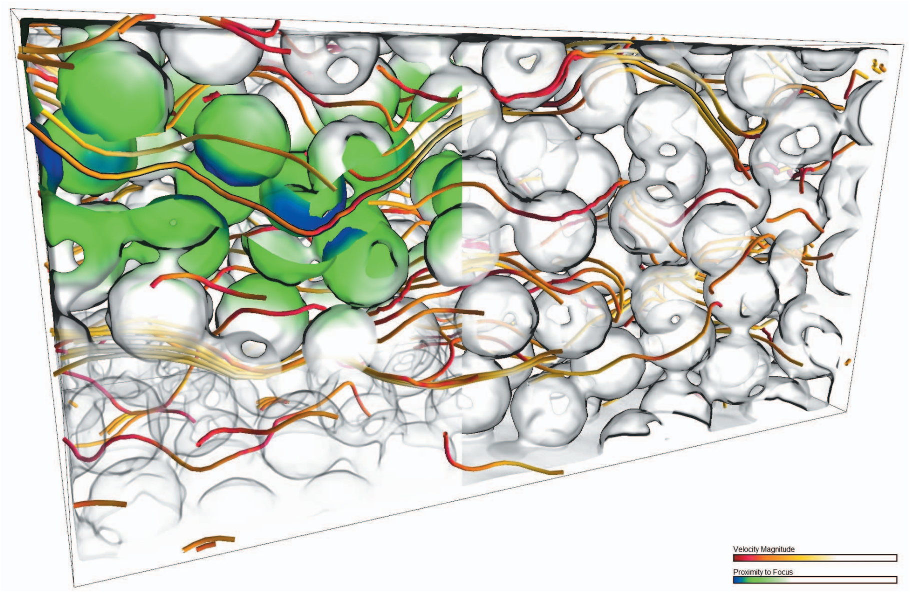


Fig. 1. Here, water particles are shown moving through a semi-porous medium of soil grains. The color along the trajectories represent speed. The left side uses our locality-based visualization to provide proximity information and focus+context for a single trajectory. The right side uses standard semi-transparent surface rendering.

Abstract—In flow simulations the behavior and properties of particle trajectories often depend on the physical geometry contained in the simulated environment. Understanding the flow in and around the geometry itself is an important part of analyzing the data. Previous work has often utilized focus+context rendering techniques, with an emphasis on showing trajectories while simplifying or illustratively rendering the physical areas. Our research instead emphasizes the local relationship between particle paths and geometry by using a projected multi-field visualization technique. The correlation between a particle path and its surrounding area is calculated on-the-fly and displayed in a non-intrusive manner. In addition, we support visual exploration and comparative analysis through the use of linked information visualization, such as manipulatable curve plots and one-on-one similarity plots. Our technique is demonstrated on particle trajectories from a groundwater simulation and a computer room airflow simulation, where the flow of particles is highly influenced by the dense geometry.

Index Terms—Flow visualization, Multi-field visualization, Focus+context visualization, Coordinated linked views.

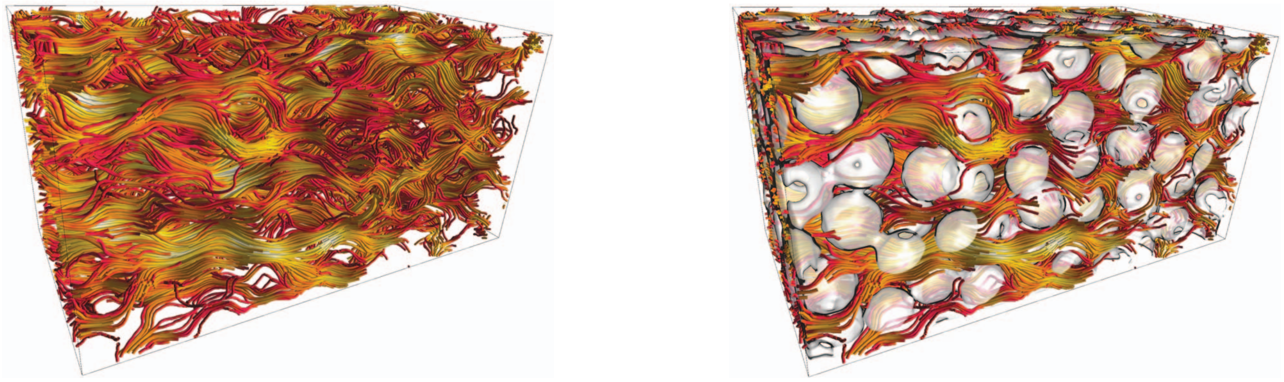
1 Introduction

The patterns and behaviors of flow, whether it be caused by gas or liquid, have intrigued and fascinated scientists for centuries. The intricate relationship between flow and the forces that direct its path create a challenging problem for scientists trying to understand and model the natural processes. Numerical simulations have advanced greatly in the last few decades, and as they grow, the field of visualization steps in to assist in rendering and analyzing the results. The most common data formats for studying flow simulations are particle and vector fields. As an example, methods such as smooth particle hydrodynamics can generate and track millions of particles moving through a contained physical space. One key feature for many areas of flow research is the use of physical barriers, such as an airplane wing or tightly packed soil grains. In these types of studies, the placement and shape of geometry directly impact the flow. Trajectories that have very similar paths may diverge after each one interacts with the physical surface differently. The relationship between a trajectory and its surrounding region provides an important avenue of research from a visualization standpoint.

The driving application for our work is the visualization of groundwater simulations. For scientists carrying out these types of simulations, which model fluids moving through porous media, the amount of solid surface accounts for a considerably large portion of the physical space, as can be seen in the images shown in Figure 2. The solid grains making up the geometry act as a barrier for the flow particles. In this single species simulation, the geometry can direct the particles to twist and turn several times as they navigate their way around the barriers. Scientists studying this effect are concerned with how prox-

- *Chad Jones is with University of California, Davis, E-mail: cejjones@ucdavis.edu.*
- *Kwan-Liu Ma is with University of California, Davis, E-mail: ma@cs.ucdavis.edu.*

Manuscript received 31 March 2010; accepted 1 August 2010; posted online 24 October 2010; mailed on 16 October 2010.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.



(a) A standard pathline visualization of the groundwater data.

(b) The pathlines with existing semi-transparent surfaces.

Fig. 2. With standard flow line visualizations, it is difficult to analyze the trajectories without first filtering data. (a) shows a large representative collection of pathlines from a groundwater simulation. The trajectories are also influenced by the surrounding solid pore grains, shown in (b). Specialized methods are needed for finding trajectories of interest and visualizing the interaction between them and the surface.

imity to the soil grains causes the direction and velocity of particle trajectories to change. By narrowing their analysis to focused subsets of the data, they can compare divergent properties between similar trajectories or study particular behaviors in greater depth. In more advanced simulations using multiple species, the geometry has an impact on where the mixing layer between species will be located, which affects how chemical reactions occur in the medium. As the numerical methods continue to advance, fluids and solids will interact directly by absorbing and changing concentrations as they come in contact with one another. In order to push the analysis process further, the visualization community needs new methods that emphasize trajectory and surface interactions.

However, understanding and comparing particle trajectories in the context of a modeled environment is not a simple task. In previous visualization systems, the flow itself takes precedence over the surrounding environment. Thus, the solid surface is commonly rendered as an opaque or semi-transparent mesh embedded in the 3D visualization. In such cases, the only information the researcher may infer is approximate locations where flow comes in contact with the object. Other visualization techniques have applied color and texturing to the surface to emphasize certain physical characteristics of the object but without showing how it relates to nearby flow.

One of the key requirements for this type of analysis process is the ability to observe and understand a trajectory's spatial behavior within the computational space. Even semi-transparent rendering techniques fail to provide clear focus+context when the geometric surface is dense and complex. In addition, properties that highlight trajectory-to-surface properties, such as local proximity, need to be effectively projected onto the 3D environment to ensure scientists can quickly determine the relationship between trajectory properties and local surface properties. Selecting a subset of data is also an essential step in creating a meaningful visualization. Existing coordinated linked views allow quantitative filtering of trajectories based on linked 2D plots. To further emphasize trajectory behavior, a 2D plot that can correlate spatial shape to other attributes would be ideal for visualizing and selecting trajectories in a different space.

Therefore, we have chosen a visualization technique we call locality-based visualization. It provides a means for understanding how a trajectory's behavior correlates to the surrounding surfaces and for visualizing locally significant properties of the flow. Along the paradigm of focused views and comparative visualization, we also provide a modification to 2D curve plots that can help illustrate divergences in trajectory values over time. To summarize, our main contributions to the field of flow trajectory visualization comes in two forms. First, we allow selection of trajectories using a linked warped curve plot that can be configured with respect to any given or derived property. Second, we display selected trajectories and the surrounding surfaces using color-mapped rendering that does not occlude the trajectories of interest. The results of these techniques are demonstrated on particle and flow simulations, namely groundwater research and computer room airflow modeling.

2 Background

Particle curves, usually streamlines, streaklines or pathlines, are the most common way to visualize 3D spatial flow. Depending on the underlying numerical method, the output from the simulation may be a vector field or a set of discrete particles, both of which usually vary over time. In addition to the spatial information, other scalar and vector values may be calculated as the simulation runs. Visualizing multidimensional data such as this has been an important topic in the visualization community for many years [20]. For an overview of recent methods for vector field visualization and integration, please refer to current state of the art reports on the subject [12, 15]. For our work, we deal with the representation and rendering of particle trajectories independent of how they are calculated, and we focus on exploratory analysis using focus+context rendering and coordinated linked views.

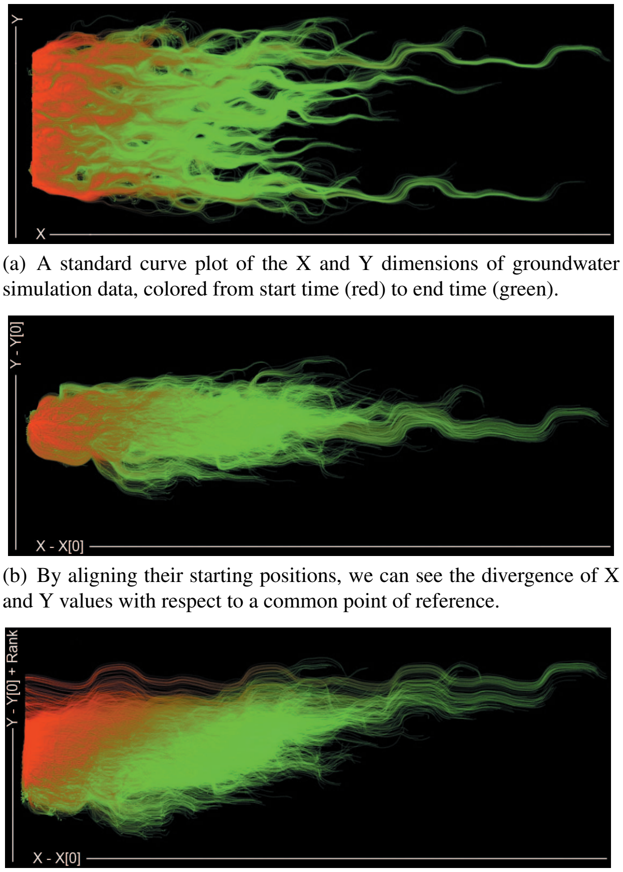
2.1 Coordinated Linked Views

Filtering trajectories using information visualization is one way to reduce the quantity of lines visible and increase the significance of the ones that are shown. Since it is difficult to capture all of the meaningful information from a multidimensional data set using a single view, multiple views are becoming common for exploring trajectory data. This method of visual analysis has been applied with much success to many different engineering problems [5]. The most common types of information visualizations used are scatterplots, parallel coordinates, and time plots.

Hurter et al. introduced FromDaDy, a simple brush, drag and drop interface for exploring 2D trajectory plots [3]. The user begins with a single 2D plot of trajectory lines using any of the available data dimensions as the axes. A new plot is generated by brushing a subset of the trajectories and dragging them onto a new location on the screen. This type of iterative interaction produces smaller and smaller plots until the desired subset of trajectories is obtained.

Instead of an iterative partitioning of trajectory plots. Willems et al. introduce a matrix of views for performing multivariate trajectory selection [19]. Attributes of the data are dragged to the rows or columns of a 2D matrix, which then produces a series of trajectory plots that match all existing pairs of attributes. The plots are coordinated such that brushing on any single plot updates the focus of all the plots.

Doleisch et al. used multiple linked information visualization interfaces for exploring their particle and flow data with their system SimVis [1, 2]. Brush filtering on scatterplots, time plots and parallel coordinates allows the user to explore the data from multiple views simultaneously. Using a mix of views means that certain features may become easier to understand in one view versus another.



(a) A standard curve plot of the X and Y dimensions of groundwater simulation data, colored from start time (red) to end time (green).

(b) By aligning their starting positions, we can see the divergence of X and Y values with respect to a common point of reference.

(c) Further manipulation of the plot allows us to sort the curves on the vertical axis according to their total displacement.

Fig. 3. A standard curve plot of the trajectories from Figure 2(a) is shown in (a). By transforming the curves, as in (b) and (c), we obtain a new view of the data. In (c) we can quickly identify the longest particle trajectories at the top of the plot. From were, we can also see that the longest trajectories have few vertical changes, which implies the particles did not need to make many turns to avoid the solid geometry.

To enhance the use of information visualization plots for pathline exploration, Shi et al. enumerated important attributes that might aid a scientist in understanding their data [16]. Developing a system similar to SimVis, the authors created an application that incorporated these attributes into the exploratory process using information visualization.

Konyha et al. used families of function graphs to help steer the visual analysis process [4]. Families of function graphs are visualized by plotting each function graph curve onto the same plot. They also incorporated other information visualizations, such as parallel coordinates, to make a complete system. Matkovic et al. continued this research topic by creating a color lines view for families of function graphs [11]. Instead of curves, they mapped each function graph to a straight line and let color denote the value of the dependent variable.

The idea of linked views is so essential that Matkovic et al. have provided a prototyping system called ComVis that allows coordinated multiple views of data to be incorporated together easily [10]. ComVis forms a basis for building new visualization systems.

Our system also takes advantage of coordinate linked views in the form attribute curve plots. By visualizing time-varying curves of particle attributes, the user can select specific particles using a simple brush interface. Going further, we introduce an adjustment to the standard plot to create warped curve plots.

2.2 Visualizing Flow with Geometry

In simulations with solid geometry, an unobtrusive context can be provided via semi-transparent rendering of the structure surrounding the flow. Some simulations also incorporate multivariate information into the structure itself, which can be shown directly on the surface using standard coloring and texturing.

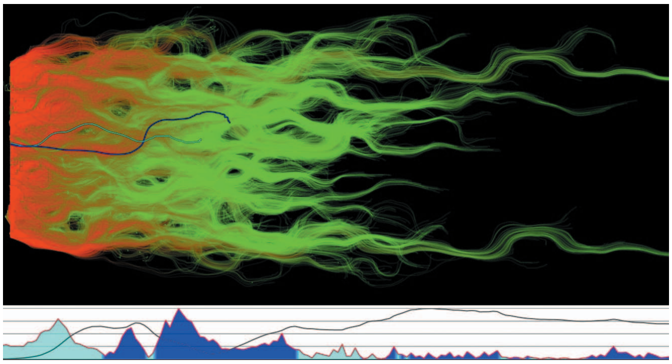


Fig. 4. The similarity signature plot, shown below the curve plot, visualizes difference between values over time for pairs of trajectories. With the area graph in this example, we can see when the velocity of one trajectory overtakes the velocity of the other by tracking the corresponding color under the area graph.

Laramee et al. visualized flow properties of a cooling jacket, where the physical geometry plays an important role in flow [7]. In addition to showing flow lines and semi-transparent rendering of the solid structure, they also displayed information on the geometry. Scalar values, such as temperature, were used in colormap rendering to denote areas of the cooling jacket with high temperature. They also applied image-space advection to identify recirculation using geometry textures [8].

Spencer et al. introduced an imaged-based approach for seeding streamlines directly on surface-based vector fields [17]. By using an image based approach, the technique is able to handle very complex geometry while maintaining high performance. The seeding strategy works automatically and ensures that streamlines on the surface are evenly placed. In similar fashion, a technique for glyph placement on vector surfaces was introduced by Peng and Laramee [14]. The glyphs are evenly spaced along the surface while allowing the user to interactively control the resolution for different regions.

The concept of limiting the 3D visualization to a local region around a trajectory has been explored in recent research. Lampe et al. introduced the technique of curve-centric reformation [6]. In their work, spatial curves through a volume were straightened, and the immediate volume around the curve was warped to match. In this way, the values surrounding the curve became easily visible while the shape of the curve was lost. Our work also focuses on showing curve-centric information, but we do so without deforming the volume or the curve.

3 Warped Curve Plots

Our research is not concerned with flow field seeding strategies, which have been covered in-depth in other research. However, we do understand the importance of limiting the amount of visible trajectories. In some cases, a single particle trajectory may be quite significant in understanding local flow behavior, and understanding a specific particle trajectory involves understanding all possible influences on its path. To this end, we provide an information visualization interface to assist in the analysis and exploratory process. We build our interface from two-dimensional curve plots to provide a way to explore multidimensional properties of pre-computed particle trajectories.

3.1 Construction

Various kinds of curve plots have also been used in other visualization systems as a selection tool [4, 10], so we will first describe the construction of a typical pathline plot. Let $L = \{l_1, l_2, \ldots l_n\}$ be the set of trajectories and $X = \{x_1, x_2, \ldots x_m\}$ be the attributes associated with each trajectory. Let the function $f_i(x_j, t)$ be the value of x_j on trajectory l_i at time t. The user selects two arbitrary attributes x_j and x_k to create a curve plot. For each trajectory l_i, an attribute curve is generated on a XY coordinate plane by plotting every data point $(f_i(x_j,t), f_i(x_k,t))$ for $t \in [0,T]$ and connecting the neighboring time steps with a line segment. A similar plot can be generated for streamlines by replacing time with a curve parameterization. Since trajectories may overlap on the 2D plane, blending is used to help keep trajectories from completely obscuring others. In other visualization systems, time is the most common choice for the horizontal axis.

We will now expand on the concept of curve plots by introducing warped curve plots. By altering the position and orientation of curves on the plot in a meaningful way, we are able to provide new views of the data. By meaningful, we mean translations and rotations that are dependent on the respective trajectory data for each curve. A constant transformation would not provide anything new.

Using the description from above, the curves on a the plot can be translated by adjusting the data points by the functions $d_1(l_i)$ and $d_2(l_i)$ such that a data point becomes $(f_i(x_j,t)+d_1(l_i), f_i(x_k,t)+d_2(l_i))$. The simplest example of a transformation is displacing the curves so that their starting points meet at the origin of the curve plot. This effectively changes the curve plot view from showing the absolute values of X and Y to showing the amount of change in X and Y over time. Thus, if two curves remain relatively close to one another, the rate of change in their respective values are similar. This type of transformation can be seen Figure 3(b). Instead of the placing the starting point at the origin, any other timestep can be moved to the origin to explicitly see changes in values from that point of reference.

The transformation can depend on any existing or derived quantity as well. For instance, each trajectory has an associated length that tells us the total displacement of the particle from start to finish. If we translate each line to the origin and then sort the lines on the vertical axis by total length, like is shown in Figure 3(c), long trajectories are separated from the short trajectories. The curves on the plot still show the relative change in X and Y, but now curves that stay close together on the plot have an extra dimension in common. This translation can be extended even further by adjusting Y in a similar manner.

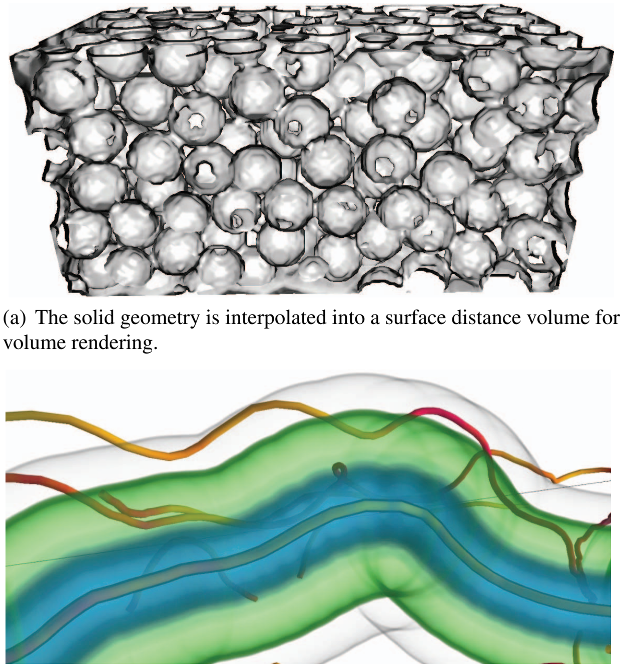
3.2 Interaction

Each curve plot that the user creates represents one more view for interactively refining a set of focus trajectories. Filtering becomes a much more important issue when the initial set of trajectories is quite large, perhaps due to the type of simulation method or a poor seeding strategy. Obtaining a set of focus trajectories allows more control over clutter reduction and locality-based visualization, which we describe in the next section. To select or deselect trajectories using the curve plots, the user simply clicks and holds the appropriate mouse button and drags the cursor across the curves on the plot. The selection persists even when the user changes or warps the axes of the plot, allowing further adjustments to the focus set.

To enhance the comparative capability of warped curve plots, we also provide a way to compare pairs of trajectories using what we call a similarity signature plot. The similarity signature plot uses area graphs to highlight the difference between two trajectories over time. The area graph in Figure 4 shows the difference in speed between the dark blue and light blue selected lines. The height of the graph represents the absolute difference in value while the color of the area shows which trajectory has the greater value. The variable used for the difference function can be changed as desired. In addition, other difference functions can be drawn on top of the area graph as normal function plots, such as the black line in Figure 4 which shows the distance between the lines at each time step. This type of in-depth quantitative analysis provides a useful addition to direct 3D visualization.

4 Locality-based Visualization

The interaction between a particle trajectory and the physical geometry plays an important role in understanding the underlying flow. We provide a technique called locality-based visualization that integrates information about a trajectory into the area surrounding the line. Focus trajectories can be selected either via 3D pick selection or via brushing the warped curve plots described in the previous section.



(a) The solid geometry is interpolated into a surface distance volume for volume rendering.

(b) A focus distance volume is created to provide locality information to the visualization system.

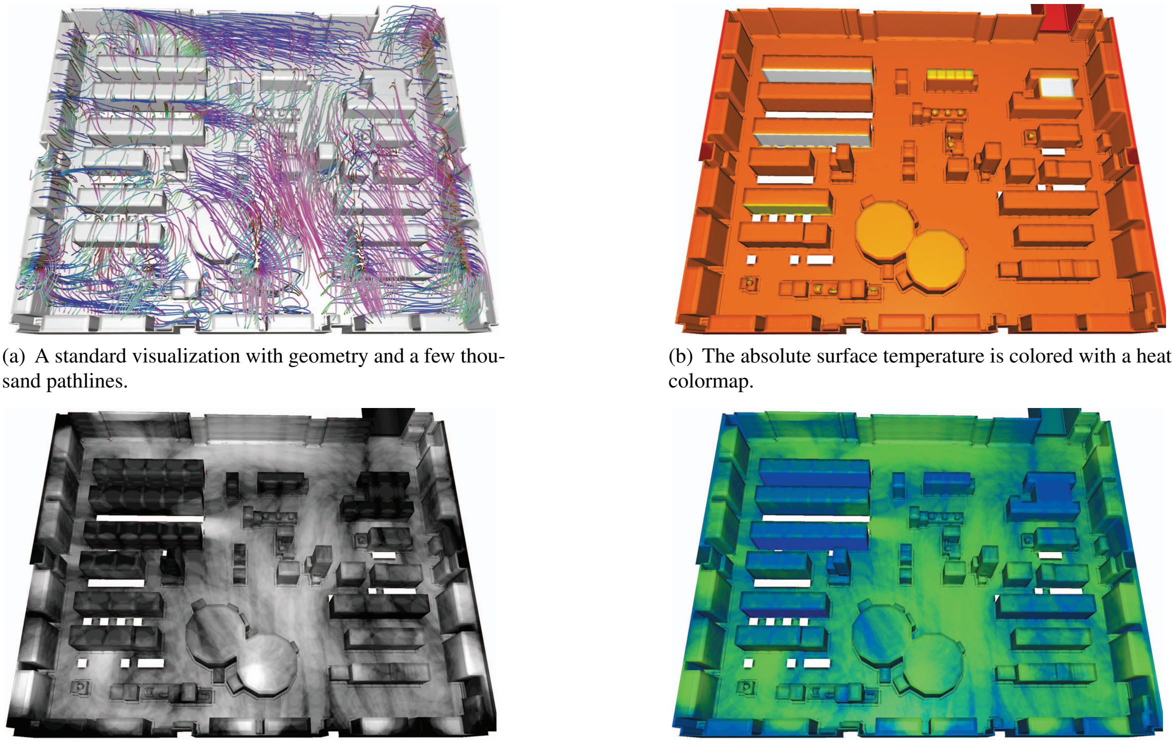
Fig. 5. We use two types of distance fields, a surface distance volume (a) and focus distance volumes (b). This combination allows us to find where trajectories overlap the solid surface and other scalar volumes.

4.1 Surface and Focus Distance Fields

The most common way of incorporating geometry into any visualization without obscuring trajectories is via semi-transparent rendering of the physical regions. It is possible to achieve this result through depth peeling of the geometry mesh or by integrating the physical regions through direct volume rendering of a 3D distance field. We chose the distance field approach for two main reasons: static particle fields can be interpolated to smooth surfaces and associated scalar fields can be directly mixed on-the-fly with the distance field. While direct surface rendering could map vertex position to texture coordinates for volume coloring, depth peeling would require multiple rendering passes, which can be costly for complex geometry. Also, by integrating the distance field into our existing GPU raycasting engine, visual enhancements, such as surface extrusion, smoothing and advanced lighting, are achievable with less effort and without direct manipulation of the geometry.

For a given physical surface, which can be represented with a geometry mesh or a particle field, we calculate a volume such that each voxel of the volume contains the distance to the nearest location on the surface. For reference purposes, we will call this volume the *surface distance volume*. Figure 5(a) shows a surface distance volume of a static particle field rendered with ambient occlusion and silhouette edge enhancements. Some fine-grain features are lost in the interpolation, but this can be alleviated by using a higher resolution volume or adaptive mesh refinement. When applicable, we try to match the size of the volume to the underly data, such as using the same dimensions of its corresponding scalar fields. If a signed distance field can not be generated, which is the case for some open surfaces, we use double-sided normals for lighting calculations.

Calculating a trajectory distance field, which we will call the *focus distance volume*, is similar to calculating a distance field to the surface. However, since the focus of our visualization can change as the user makes new selections, the focus distance volume must be recomputed quite often on-the-fly. Therefore, we utilize the power of the GPU by calculating the focus distance volume using CUDA. In the CUDA kernel, each thread is responsible for a single voxel's distance calculation.



(a) A standard visualization with geometry and a few thousand pathlines.

(b) The absolute surface temperature is colored with a heat colormap.

(c) The logarithmic frequency in which pathlines pass near the geometry.

(d) Combining the two results lets us show the correlation between temperature and locality frequency.

Fig. 6. The input to this visualization is a set of uniformly seeded pathlines, shown in (a). One of the simulation scalar volumes, temperature, is shown on the surface of the geometry in (b). The heat colormap shows that the highest amount of heat, which appear as white and yellow, is located on the disk archivers and computer clusters. In CUDA, we compute a proximity frequency volume that counts the frequency of trajectories passing near each voxel, the result of which is shown in (c). By combining both results into a correlation volume in (d), we can see areas of high temperature and low frequency (blue) and areas of low temperature and high frequency (green).

For a very small number of lines, it is faster to use a kernel that simply calculates the distance from the voxel to every line segment of the focus set. For larger focus sets, we use a uniform grid data structure that contains ids to each line segment within that region. Thus, the thread need only to look in a limited number of surrounding voxels to compute a distance. The CUDA kernel could be speed up even more by using a more advanced data structure, such as a kd-tree or an octree, which allow for nearest point searches.

4.2 Focus+Context Rendering

Rendering is performed in two passes. The first pass renders all of the particle trajectories as lit tubes. These are stored in a framebuffer object with color and depth information. In most cases, this is the slowest pass since a large collection of trajectories may generate hundreds of thousands to millions of triangles. Of course, any type of line or tube rendering approach, such as illuminated lines or view-aligned impostor tubes, could also be used to prevent slowdown with larger sets. In the second pass, we render the distance field and any scalar volumes that the user would like to incorporate into the visualization. By using raycasting for the volume rendering pass, we can accurately place the framebuffer color for the trajectories at the proper depth location.

Once we have both the surface and focus distance volumes, we can also provide a focus+context view. Using direct volume rendering for the surface distance volume allows us to provide semi-transparent views with illustrative techniques such as edge enhancement. If we combine this rendering with the focus distance volume, it is then possible to further modulate opacity to reduce clutter in regions that are not directly surrounding the focus items. This means only regions near the trajectories of interest are fully shown, as can be seen in Figure 1.

Sometimes the surface geometry can still obscure the view of the focus trajectories if it becomes between the viewer and curves. Therefore, we provide an optional view-aligned cutaway that reveals the hidden curve. For every focus trajectory, we generate an expanded, view-aligned quad strip. Using a second framebuffer object, we capture the depth of the expanded curves. In the volume renderer, the opacity of the ray is reduced if that ray would intersect a segment of the expanded curve, thus providing a cutaway view. Examples of this effect can be seen in Figure 1, where the particle trajectories would normally be heavily obstructed by the soil surfaces.

4.3 Projecting Data Values

Now that we have a focus+context view that is aware of the local focus regions, we can project specific quantities onto the solid surface near our focus. The simplest information that can be projected onto the surface is proximity itself. Specifically, the color and opacity along the surface represent how close the focus trajectories pass by the solid geometry. This allows a line to be colored by another property, such as velocity magnitude. Thus, identifying correlations between surface proximity and the trajectory's attributes becomes an easier task. When dealing with multiple focus lines, we can either visualize minimum proximity or distance-based frequency. With distance-based frequency, as shown in Figure 6(c), the method calculates how many of the trajectories pass by the surface within a given distance r. Visualizing frequency count on the surface is useful when the quantity of pathlines at a specific location is important, such as finding which soil granules of a groundwater simulation encounter the highest number of fluid particles. For vector fields, it is important to note that particle seeding has a significant impact on frequency counts, and thus, the calculation should be scaled to account for the number of trajectories.

We can still color the local surface by the other scalar or particle fields, such as temperature or pressure, but we can also generate new scalar fields to represent differences between the focus values and the absolute values. When dealing with dense, unstructured particle fields, we can interpolate the particles on a uniform grid using weighted averaging or moving least squares. We utilize CUDA and GPU hardware to speed up the calculation, similar to [9].

5 Results

We applied our technique to two datasets: a computer room airflow simulation and a groundwater simulation. In terms of rendering cost, the largest bottleneck is usually the trajectory geometry, as we discussed in the previous section. The additional cost of rendering the focus distance volume with the surface distance volume is an extra texture lookup and a few shader operations. For the groundwater simulation, which is over 14 million voxels, our focus+context technique took the average rendering rate down from 19 fps to 15 fps at 1020x768 resolution with a constant sample spacing, and most of the extra cost was due to the lowered opacity causing fewer fragments to terminate early. Table 1 shows average seconds required to compute a focus distance volume based per thousands of line segments (LS), where a typical trajectory consists of around 100 segments. Our future work will include optimizing the CUDA code to handle much larger focus sets at faster speeds.

Table 1. Timings on Nvidia GTX 260 for focus distance volume generation in seconds per thousands of line segments (LS).

Data	Volume Size	1k LS	10k LS	100k LS
Ground	384x192x192	0.36 secs	1.53 secs	45.44 secs
Airflow	420x345x60	0.14 secs	0.97 secs	31.02 secs

5.1 Computer Room Airflow Simulation

The following results show images of an airflow simulation for the RIKEN Super Combined Cluster (RSCC). The goal of the simulation is to understand the thermal flow of the room and enhance cooling efficiency. This, however, depends on the position of the machines in relation to the flow of air. The computers and storage machines in the room provide heat sources while the air conditioners along the walls provide airflow and cooling.

For our case study, we uniformly seed pathlines throughout the simulation space. To understand the effects of circulation in relation to temperature, we generate a volume that counts the frequency of unique pathlines passing near each voxel and combine that with the surface distance field. As stated in the previous section, localized pathline frequency is highly dependent on the quantity and placement of seed points, and thus, we attempt to saturate the space to ensure a dense collection of circulating trajectories. The pathlines can be seen in Figure 6(a). The temperature on the surface colored using a heat colormap is shown in Figure 6(b), and the frequency of pathlines passing within 8 voxels away from the surface is shown in Figure 6(c). If we apply a correlation function on these two volumes and project the values onto the surface, as in Figure 6(d), we can see negative correlations as blue and green. Blue regions represent areas where temperature is much higher relative to frequency while green regions represent areas where temperature is much lower relative to frequency. Therefore, we can make some assertions about how the flow is affecting the temperature of these heat sources. Namely, the machines that are not receiving a high frequency of the particle trajectories are much higher in temperature. With corrections to the layout of fans, circulation could be improved to help funnel air to regions where it is needed most.

In another example, we look at the distribution of trajectories in relation to their velocity. The computer room contains rows of air conditioners along the walls, and these units pull in hot air from the top of the simulation and pump cool air out the bottom. One way to see the efficiency of these machines is to check the velocity magnitude in relation to the distance from the floor. If the trajectories are moving quickly near the ceiling and near the floor, then the air conditioners' throughput is good. We first plot the trajectories on the curve plot to associate height (vertical) and velocity (horizontal), as shown in Figure 5.1. After brushing the appropriate curves, we then use a proximity coloring on the geometry to find areas where the fast moving trajectories pass close by the surface. Some of the fastest trajectories are located in the corner of the machine room. This shows air is being pumped into the unit quickly, allowing for fast cooling.

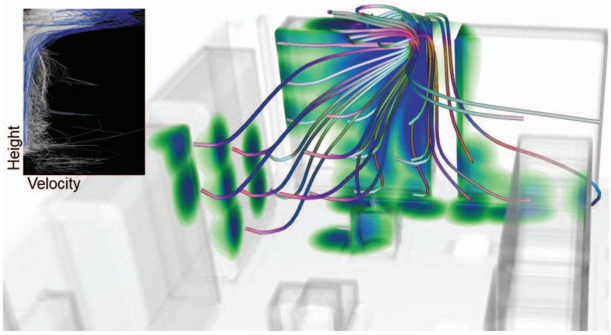


Fig. 7. The curve plot and spatial view shown above reveal how the air conditioners pull in warm air near the ceiling so that it can cool and recirculate it back into the room through the floor.

5.2 Groundwater Simulation

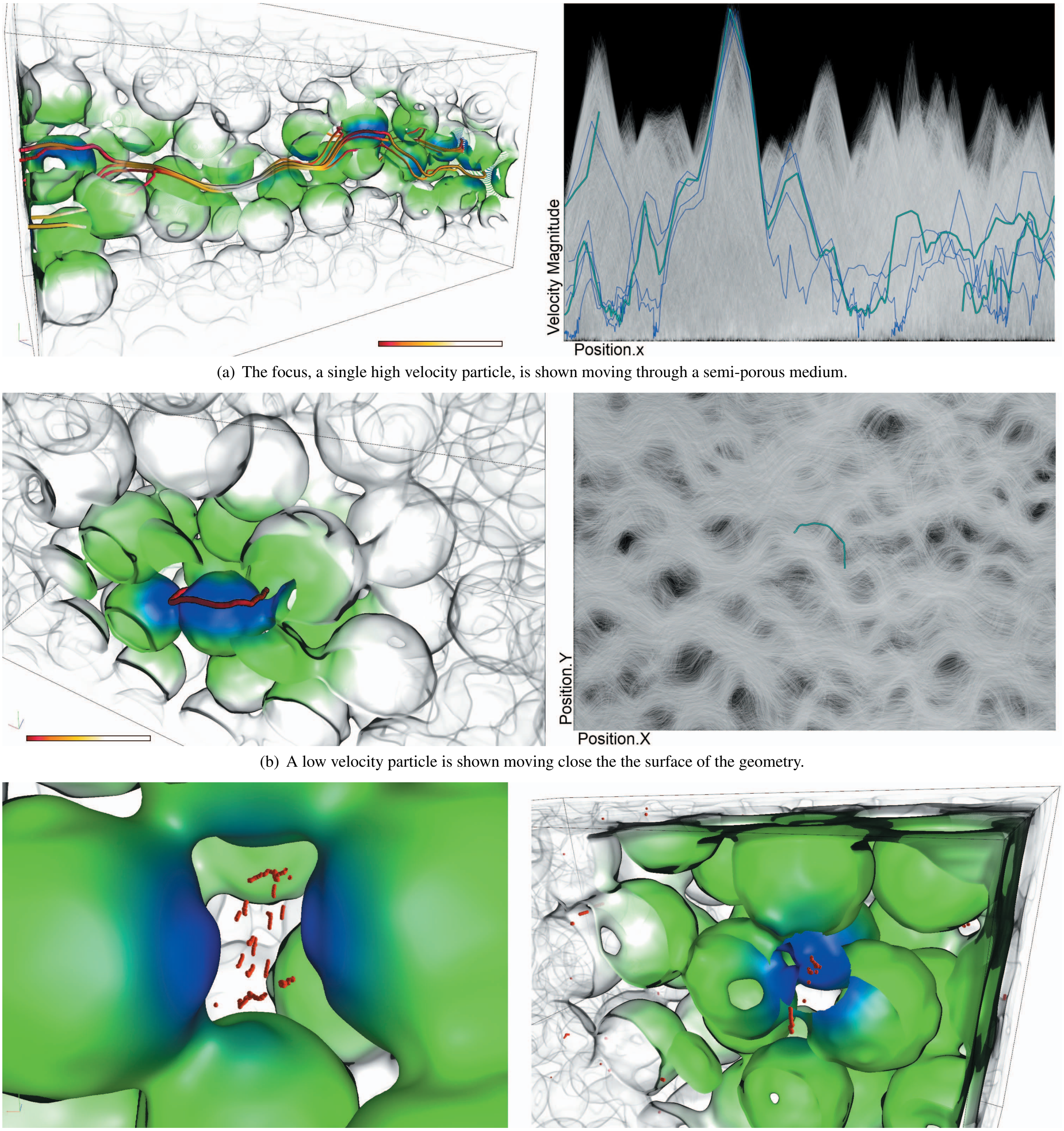
The groundwater scientists at Pacific Northwest National Lab perform simulations to model flow and solute transport in porous media at the pore-scale, in which the small, discrete geometry of solid grains and pore spaces is explicitly resolved. They use a fully lagrangian particle-based numerical method called smoothed particle hydrodynamics [18] to solve the Navier-Stokes equations of flow and the advection and diffusion of dilute solutes. Their code is implemented on high-performance parallel computing systems and scales efficiently to thousands of processor cores [13]. The simulation data shown here was generated from 14 million particles that represent the fluid and solid phases. The most important aspects of these early studies deal with discovering and understanding pathways of high and low velocity in relation to the physical pore granules.

The following examples examine the correspondence between particle velocity and proximity to a solid surface. In these examples, pathlines are traversed for 150 time steps and are colored by velocity magnitude using a heat colormap. In Figure 8(a), we select a set of trajectories by brushing the curve plot of velocity magnitude (vertical) versus X position (horizontal). We select a specific peak along the X dimension which exhibits the largest velocity, and visualize those trajectories in physical space. The peak velocity portion of the trajectories is shown in white along the tube. We use a mouse pick to focus on a single trajectory out of the set and generate its respective focus distance field, which is used to provide a focus+context fading of the surrounding solid surface. We then apply a colormap to the nearby regions that shows when the trajectory passes near (blue) or far (green) from that surface. We can identify that when the trajectory has high velocity, the surface surrounding the trajectory is green. Conversely, in areas where the surface is blue, the trajectory exhibits a lower velocity.

The next two figures tell a different story about trajectories and velocity. In Figure 8(b), we have highlight a single trajectory that has a near constant low velocity over the time sequence. The pathline was chosen from a warped curve plot similar to the one shown in Figure 3(c) by brushing a small displacement curve. We then switch the curve plot to show the X (horizontal) and Y (vertical) position of the trajectory in space. Our locality-based visualization shows that the trajectory traverses the physical space in close proximity to the surface during the entire time sequence. This illustrates the impact that surface proximity has on flow velocity. Figure 8(c) shows an extreme case of flow stagnation by visualizing two immobile zones, where trajectories have the lowest possible total displacement. Each zone contains clusters of trajectories that have very low overall velocity, and the pocket of stagnant flow becomes easily visible using our view-aligned cutaway.

5.2.1 Scientist Comments

Understanding small scale models, shown in this paper at the pore-scale, allows us to improve our understanding of fluid flow at larger scales. In scales where pore-scale geometry is not explicitly considered, the effects of pore-scale variations are often represented by a



(a) The focus, a single high velocity particle, is shown moving through a semi-porous medium.

(b) A low velocity particle is shown moving close the the surface of the geometry.

(c) Two immobile zones of flow are shown with with low displacement, low velocity particles.

Fig. 8. Here, we show a set of results obtained from using our visualization techniques on a simulation generated by groundwater scientists at Pacific Northwest National Lab. In (a) we have selected a set of trajectories on the curve plot that reach a high velocity peak at a specific location along the X dimension. In the spatial view, we color trajectories based on velocity magnitude using a heat colormap. After focusing on a single trajectory with the mouse, we generate a locality-based view around the trajectory to emphasize regions where the trajectory passes far (green) and near (blue) the surface. The high velocity of the trajectory correlates to green regions. The example in (b) shows the opposite example, where a trajectory has a low velocity and stays consistently near the surface. The most extreme example of stagnant flow is shown in (c).

macroscopic process known as dispersion, which is typically modeled as a Fickian process. However, certain velocity correlations or pockets of relatively immobile water can lead to non-Fickian behavior, which manifest at large scales as preferential flow pathways or long-tailed arrival time distributions. We use visualization tools to examine the pore-scale velocity distributions generated by the SPH simulations to identify features that might given rise to the types of phenomena described. Many widely-available visualization packages, however, are oriented toward grid-based data rather than particle and trajectory data.

The tools shown here are well-suited for time-varying particle visualization and have been adapted for the specific purpose described above. The basic visualization provides semi-transparent surface rendering of solid grains and colored geometry rendering. Selected points or pathlines can be animated over time to visualize the pore-scale transport processes. These visualizations provide an engaging and illuminating visual representation of this process. By filtering with curve plots, selected pathlines are combined with color-coded solid surface rendering to allow visual identification of where selected particles are close to or far from a solid surface. This is important because the solid surface is represented as a zero-velocity boundary, and therefore, the proximity to solid faces is closely related to local velocity.

Thus, particles that spend an extended time period near granule faces contribute to long-tailed arrival distributions, and conversely, those that remain near pore centers represent preferential flow pathways and lead to early arrival. By studying the extracted time series, we can define temporal correlation structures which will provide information needed to develop alternatives to the traditional Fickian dispersion model. The visualization has also identified some particle sets that remain ”trapped” in nearly immobile zones formed where the geometry causes stagnation of flow. Our research relies on analyzing the volumetric extent of such immobile zones in order to understand the physical basis for porosity fractions and quantifying the potential for mass transfer between mobile and immobile zones.

Our model was recently enhanced to include diffusion between fluid and solid phases and will soon incorporate chemical reactions between multiple solutes. Future visualizations of these processes will provide insight into the relationship between pore-scale intragranular diffusion and macroscopic multi-rate mass transfer models. We can then better analyze the effect of pore-scale variations in velocity on solute mixing and the control exerted on the extent of chemical reactions.

6 Conclusion

We have presented a technique for exploring the local interaction between flow trajectories and nearby solid surfaces. This is enhanced by linked information visualizations in the form of warped curve plots. The exploration process begins with a collection of pre-generated trajectories from a flow field simulation. Warped curve plots let the user interactively select which trajectories to view in physical space by brushing curves generated by pairs of attributes. The curve plots can be modified using other attributes to help structure the layout of curves and provide a way to see how similar curves group together or diverge. In physical space, a locality-based visualization technique is used to provide focus+context rendering and emphasize proximity information between trajectories and the solid surface. The approach was applied to two different simulations: understanding the correlation between airflow and surface temperature in a computer room simulation and understanding the behavior of fluid particles as they move through a semi-porous medium in a groundwater simulation.

Traditional flow visualization methods allow scientists look at their data directly with tracer particles and surface or volume rendering. As scientists begin to simulate more complex phenomena using increasingly more powerful supercomputers, they will need new analysis and visualization tools to cope with the extra complexity. Our method helps scientists to look at their flow fields from a different perspective in space and time, leading to a better understanding of trends and causal relationships in the data. We hope to see techniques similar to ours in the next-generation visualization tools.

Acknowledgments

We would like to thank Tim Schiebe and Bruce Palmer at Pacific Northwest National Lab for providing simulation data and feedback. This research was supported in part by the U.S. National Science Foundation through grants OCI-0749227, OCI-0749217, OCI-00905008, OCI-0850566, and OCI-0325934, and the U.S. Department of Energy through the SciDAC program with Agreement No. DE-FC02-06ER25777 and DE-FG02-08ER54956.

References

[1] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *VisSym*, 2003.

[2] H. Doleisch, M. Mayer, M. Gasser, P. Priesching, and H. Hauser. Interactive feature specification for simulation data on time-varying grids. In *SimVis*, pages 291–304, 2005.

[3] C. Hurter, B. Tissoires, and S. Conversy. Fromdady: Spreading aircraft trajectories across views to support iterative queries. *IEEE Transactions on Visualization and Computer Graphics*, 15:1017–1024, 2009.

[4] Z. Konyha, K. Matkovic, D. Gracanin, M. Jelovic, and H. Hauser. Interactive visual analysis of families of function graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12:1373–1385, 2006.

[5] Z. Konyha, K. Matkovic, and H. Hauser. Interactive visual analysis in engineering: A survey. In *Spring Conference on Computer Graphics*, pages 31–38, 2009.

[6] O. D. Lampe, C. Correa, K.-L. Ma, and H. Hauser. Curve-centric volume reformation for comparative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15:1235–1242, 2009.

[7] 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. *IEEE Visualization Conference*, 0:79, 2005.

[8] R. S. Laramee, J. J. van Wijk, B. Jobard, and H. Hauser. Isa and ibfvs: Image space-based visualization of flow on surfaces. *IEEE Transactions on Visualization and Computer Graphic*, 10(6):637–648, 2004.

[9] C. Ledergerber, G. Guennebaud, M. Meyer, M. Bacher, and H. Pfister. Volume mls ray casting. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1546, 2008.

[10] K. Matkovic, W. Freiler, D. Gracanin, and H. Hauser. Comvis: a coordinated multiple views system for prototyping new visualization technology. In *International Conference Information Visualisation*, 2008.

[11] K. Matkovic, D. Gracanin, Z. Konyha, and H. Hauser. Color lines view: An approach to visualization of families of function graphs. *International Conference on Information Visualization*, pages 59–64, 2007.

[12] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen. Over Two Decades of Integration-Based, Geometric Flow Visualization. In *EuroGraphics*, pages 73–92, 2009.

[13] B. Palmer, V. Gurumoorthi, A. M. Tartakovsky, and T. D. Scheibe. A component-based framework for smoothed particle hydrodynamics simulations of reactive fluid flow in porous media. *International Journal of High Performance Computing Applications*, 2010.

[14] Z. Peng and R. S. Laramee. Vector glyphs for surfaces: A fast and simple glyph placement algorithm for adaptive resolution meshes. In *Proceedings of Vision, Modeling, and Visualization*, pages 61–70, 2008.

[15] Z. Peng and R. S. Laramee. Higher Dimensional Vector Field Visualization: A Survey. In *Theory and Practice of Computer Graphics*, 2009.

[16] K. Shi, H. Theisel, H. Hauser, T. Weinkauf, K. Matkovic, H.-C. Hege, and H.-P. Seidel. Path line attributes - an information visualization approach to analyzing the dynamic behavior of 3D time-dependent flow fields. In *Topology-Based Methods in Visualization II*, pages 75–88, 2009.

[17] B. Spencer, R. S. Laramee, G. Chen, and E. Zhang. Evenly spaced streamlines for surfaces: An image-based approach. *Computer Graphics Forum*, 28(6):1618–1631, 2009.

[18] A. M. Tartakovsky, P. Meakin, T. D. Scheibe, and B. D. Wood. A smoothed particle hydrodynamics model for reactive transport and mineral precipitation in porous and fractured porous media. *Water Resources Research*, 43(5), 2007.

[19] N. Willems, H. van de Wetering, and J. J. van Wijk. Multivariate trajectory selection. *Workshop on Behavior Monitoring and Interpretation*, 2009.

[20] P. C. Wong and R. D. Bergeron. 30 years of multidimensional multivariate visualization. In *Scientific Visualization, Overviews, Methodologies, and Techniques*, pages 3–33. IEEE Computer Society, 1997.