Scalable Visualization of Discrete Velocity Decompositions Using Spatially Organized Histograms

Tyson Neuroth* UC Davis
Franz Sauer† UC Davis
Weixing Wang ‡ Princeton Plasma Physics Lab
Stephane Ethier § Princeton Plasma Physics Lab
Kwan-Liu Ma¶ UC Davis

ABSTRACT

Visualizing the velocity decomposition of a group of objects has applications to many studied data types, such as Lagrangian-based flow data or geospatial movement data. Traditional visualization techniques are often subject to a trade-off between visual clutter and loss of detail, especially in a large scale setting. The use of 2D velocity histograms can alleviate these issues. While they have been used throughout domain specific areas on a basic level, there has been very little work in the visualization community on leveraging them to perform more advanced visualization tasks. In this work, we develop an interactive system which utilizes velocity histograms to visualize the velocity decomposition of a group of objects. In addition, we extend our tool to utilize two schemes for histogram generation: an on-the-fly sampling scheme as well as an in situ scheme to maintain interactivity in extreme scale applications.

Keywords: 2D histogram, velocity decomposition, particle data, movement data, large-scale data.

1 INTRODUCTION

The ability to visualize the motion of a group of objects has applications to many fields. For example, the study of fluids often uses large sets of tracer particles to represent the intricate flow patterns in both a laminar and turbulent setting. Presenting the motions of these particles in a coherent fashion can lead to a better understanding of the underlying processes involved in phenomena such as fusion [1, 16] or combustion [18]. Geospatial movement data is another area which focuses on detecting motion-based patterns of objects such as animals or motor vehicles. For example, studying trends in the movement of marine life [2] can lead to a better understanding of migration or feeding patterns. These are simply a few examples as there are many areas of study that focus on the motional behavior of a group of objects.

Visualization techniques are often subject to a trade off between clutter and loss of information. For example, visualizing the velocity decomposition of a group of objects using vector plots can result in a great deal of clutter. Averaging the motion into a velocity vector field can alleviate this problem, however significant information about the true underlying velocity decompositions may be lost. This paper focuses on developing a visualization tool that can address these issues.

Velocity-based histograms work by sampling the frequency of objects that exhibit a particular velocity decomposition. In our implementation, each histogram bin is represented by a cell in a 2D grid and it's associated frequency is visualized using a color map. This results in an information dense visualization of the overall motion of a group of objects, with a nice regular structure that lends well to user interaction and efficient computation. Many of the challenges of this technique lie in the ability to present many velocity histograms to users simultaneously (each representing a different portion of the domain).

Another challenge lies in visualizing large scale datasets where both computational and perceptual scalability become a concern. Due to the many parameters that can be chosen when generating histograms to highlight different aspects of the data, this technique is best suited as an interactive visualization tool. However, special care must be taken in order to maintain real time histogram generation in large scale datasets. As a result, we develop both an on-the-fly and in situ method of histogram generation to handle a large variety of dataset types and sizes.

In this paper, we present our histogram-based visualization techniques and make the following contributions:

- We apply the velocity histogram to generate a low-clutter visualization of the motion of a group of objects.
- We develop two techniques for histogram generation: an on-the-fly sampling scheme for smaller datasets and an in situ scheme for large-scale datasets.
- We demonstrate an interactive technique for relating the histograms to the raw particle data.
- We implement a set of interactive tools which can be used to explore both small and large-scale datasets in real time.
- We integrate these techniques into a coherent and usable visualization software.

We demonstrate the effectiveness of our approach in many applications using a number of fundamentally different datasets.

2 RELATED WORK

There are a number of works that focus on reducing clutter in flow visualization that are applicable to this problem since many flow datasets can take a Lagrangian (particle-based) representation. Kirby and Laidlaw [8] use concepts from painting to design a multilayer representation of colors and textured patterns to represent different parameters of a flow field in an easy to read manner. Obermaier and Joy [10] use a unique visualization of metric tensors to show deformations on 3D surfaces in a flow. Properties like the velocity gradient can be represented on these surfaces using elliptical glyphs. In addition, topologically-based methods, which were first introduced by Helman and Hesselink [4], can be used to extract specific flow patterns of interest and present them in a low-clutter way.

Many of these methods still suffer from the fact that they can hide useful details in the system of study. For example, information from a stray tracer particle moving against the majority of the flow is likely to become lost entirely as major trends will always be favored over subtle ones. In addition, many of these techniques have been designed specifically for flow visualization applications. A general visualization system that is designed to work well in many different applications is often desirable.

Representing motion in the histogram-based format has made a stronger presence in computer vision than it has in flow visualization. For example, Ihaddadene and Djeraba [6] utilize a block direction histogram to represent the overall motion of crowds in differ-

*e-mail: taneuroth@ucdavis.edu
†e-mail: fasauer@ucdavis.edu
‡e-mail: wwang@pppl.gov
§e-mail: ethier@pppl.gov
¶e-mail: ma@cs.ucdavis.edu

ent "blocks" or fields of view. Romanoni et al. [12] utilize spatio-temporal histograms for background subtraction in cases where the camera is in motion. Another example is the use of "Histograms of Oriented Gradients" as feature descriptors. Dalal et al. [3] utilize these for the accurate detection of humans that are in motion relative to the background or camera. Lastly, velocity histograms were utilized by Jung et al. [7] in the clustering of visually tracked objects. The clusters are identified based on the motion of groups of objects by identifying similar neighborhoods in the resulting velocity histograms.

The sampling window design and histogram overlay used in this work to help support interactive histogram based selection, exploration, and visualization can be considered a type of "Magic Window", as defined by Tominski [15]. In addition, glyphs have been designed for visualizing uncertain flow fields by showing ranges of possible velocities [17]. Hlawatsch et. al [5] have designed a glyph which also encodes statistical information and utilizes a color map.

Storing in situ generated distributions as a way of downscaling the data storage of results from large scale simulations is not uncommon. Thompson et al. [14] leverage such in situ generated histograms in a volume rendering application utilizing these histograms for feature detection through statistical and topological methods.

Velocity histograms in the form utilized in this work have, for the most part, only been used on a basic level throughout the domain specific community. They have been shown to be useful in the areas of geophysics [9], superfluids [11], and fusion [13] to name a few. There has been very little work in the visualization community on leveraging them to perform more advanced visualization tasks or implementing them into an interactive system.

3 Methods

The main application of this technique is for interactive visualization of the velocity decomposition of a group of objects. The velocity of each object is sampled in order to form a set of spatially organized velocity histograms to be visualized by the user. The means of sampling and generating the histograms, as well as their visual appearance and spatial layout, are all important factors that must be considered and are described in more detail in the following sections.

3.1 Overview

One of the benefits of our system is the ability to interactively explore the data in real time with various adjustable parameters and configurations. For these reasons, the histograms must be computed at runtime. The binning process is the most computationally intensive step and can be done efficiently using GPU acceleration. For large scale datasets, we provide an alternative method which can emulate the interactivity of forming histograms on-the-fly by sampling from a set of high resolution histograms that are constructed in situ. Figure 1 shows an overview of the steps in our workflow with the original scheme highlighted in blue and the in situ scheme highlighted in green.

3.2 Velocity Histograms

Figure 2 shows a comparison of the velocity histogram to two other techniques that attempt to visualize the motion of a group of objects. The most straightforward method, shown in the left of the figure, involves adding a velocity vector to each object. The direction and size of the arrow can be used to infer the speed and direction of motion of that particular object. The downside to this technique is that it can result in a great deal of clutter when the number of objects is large, making it difficult to visualize patterns. An alternative to this technique, shown in the middle of the figure, is to average the velocities of all objects into a vector field. This has the advantage of reducing the amount of clutter, but also results in a loss of information about the underlying velocity decomposition. Another option is to use trajectories of pathlines. However, this can once again result in clutter and becomes difficult to indicate which way along the pathline the object is moving. Furthermore, there is some temporal confusion since a drawn trajectory represents multiple points in time.

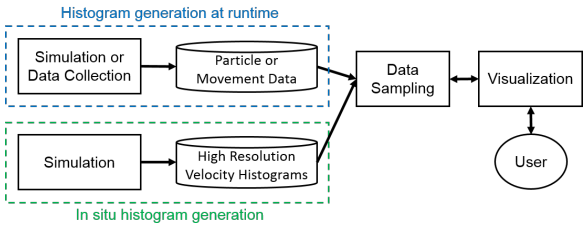

Figure 1: An overview of our workflow showing two different schemes. Blue) Raw particle or movement data is saved and sampled into velocity histograms at run time. Green) High resolution histograms are generated using the full particle data in situ and are then sampled to generate the visualization.

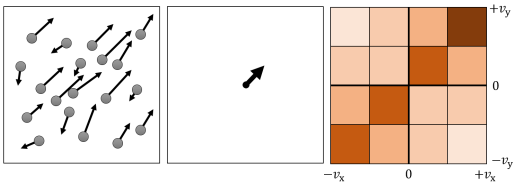

Figure 2: An image comparing techniques of visualizing the velocity decomposition of a group of objects. Left) Assigning a vector to each object. Middle) Average the velocities into a vector field. Right) Sampling into a velocity histogram with darker colors indicating a higher frequency.

This paper focuses on the use of velocity histograms, which is shown in the right of the figure. The velocity histogram has the advantage of being able to accurately show quantitative information about a large number of objects in a low-clutter, well organized visualization which supports easy, interactive quantitative exploration. The velocity of each object is sampled and binned into an appropriate histogram cell. Color can then be used to indicate the frequency of objects occupying each cell. In the example shown in the figure, it is clear that the majority of objects have a velocity with a large positive x and y component (as indicated by the dark cell at the top-right of the histogram). On the other hand, few to no objects are moving to the bottom-right or top-left (as indicated by the lightly colored cells in those regions).

Such a histogram is a powerful method of visualizing overall trends of the motion of a group of objects. We can then further extend this technique to also study the differences in the motion in various parts of the domain. This is done by decomposing the domain into a desired number of subsections, each with its own velocity histogram. All the objects within a particular subsection are then sampled to form a velocity histogram for that region. A comparison between the histograms in different parts of the domain can show spatial variations in motion of the objects. Moreover, adjusting the resolution of this domain decomposition can highlight general trends that permeate larger portions of the domain, as well as small-scale details. This is discussed further in Section 3.3.

Figure 3 shows an example of a synthetic particle dataset designed to highlight the advantage of the velocity histogram-based view. It consists of $\sim 200,000$ randomly distributed particles whose velocity in the x-direction (horizontal) is a positive random number and increases with the x position for some particles. The velocity

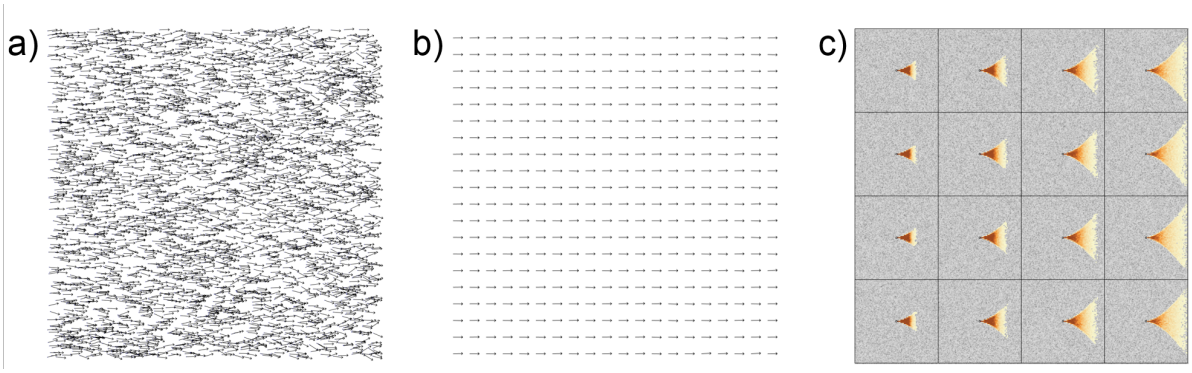

Figure 3: A synthetic particle dataset designed to show the advantages of using the histogram based representation. a) Attaching a velocity vector to each particle results in a great deal of clutter. b) Averaging particle velocities into a vector field eliminates important flow patterns. c) The histogram based view provides a low clutter decomposition of the velocities in each part of the domain.

in the y-direction (vertical) is a positive or negative random number which increases with x position for all particles. The y-direction velocity has an equal chance of being positive or negative. The fact that many particles are moving in opposite directions throughout the domain helps demonstrate the advantage of the histogram-based view: a) An arrow is attached to each particle to indicate its velocity. Drawing only 2% of the full data still results in an abundance of clutter. b) Averaging the particle velocities into a vector field causes the opposite motions of particles in the y-direction to be averaged out and become lost. Moreover, the increase in velocity in the x-direction is also difficult to notice since only a few particles display this variation. c) In the histogram-based view, each histogram highlights the overall decompositions of velocities in each portion of the domain. Darker colors represent a higher frequency of particles. The center of each histogram represents no movement whereas a highlighted bin towards the top-right of the histogram, for example, would represent fast motion in the positive x and y directions. Since the histograms represent distributions, many of the subtle patterns that would normally becomes lost via averaging are easily seen.

Flows like the one used in this example, where particles move differently from one another, even when occupying the same location in the domain, are commonly found in certain applications, such as fusion science. The plasmas that flow through fusion reactors contain oppositely charged particles which behave uniquely from one another in the presence of an electromagnetic field. This makes many traditional visualization techniques limited. The histograms used in our visualization tool are 2D in nature. This is because occlusion makes visually representing a 3D histogram very difficult. As a result we employ a number of interactive techniques so that users can explore 3D domains to a limited extent. This is discussed in more detail in Section 3.3.

3.2.1 Generation from Particle Data with GPU Acceleration

In our algorithm, the texels of a single channel texture store the values of the histogram bins. The object velocities and positions are transferred to the GPU in the form of a vertex buffer. Interactive parameters, such as the position and structure of the sampling grid, the histogram resolution, and the normalization factors, are transfered to the GPU as well. The vertex shader then maps each object to a histogram bin based on it's position and velocity. The result is a texel index which must then be converted into the correct position in normalized device coordinates before being input to the fragment shader. With additive blending enabled, the fragment shader then increases the value of the texel corresponding to the mapped histogram bin. The end result is a texture storing the computed histograms that can be queried as necessary to support interactive features before being rendered to the screen using an adjustable transfer function.

3.2.2 In Situ Generation and Sampling

While the on-the-fly sampling scheme is easily manageable for medium and small-scale datasets, at larger scales computational overhead limits the interactivity of the system. As a result, we also implement an in situ scheme which can visualize velocity decompositions in extreme-scale applications without losing interactivity. In this scheme, particles are directly sampled and transformed into a histogram-based representation during the runtime of the simulation allowing us to accumulate the statistical information of objects at much larger scales.

Each histogram is sampled onto a mesh, dependent on the type of simulation, where each grid point represents the center of a 3D volume from which objects are sampled. Since the size of each sample region can vary, especially in unstructured grids, the volume of the sample area is saved in addition to the accumulated high-resolution histogram. This will in turn be used to more accurately sample histograms in the visualization tool. The histograms themselves represent distribution functions which are normalized using a normalization factor according to their relative intensities and are saved at key timesteps throughout the simulation.

Our interactive system then samples this more manageable data representation for interactive exploration. Because the grid sizes can be dense and unstructured, it is often effective to visualize an overview at a reduced level of detail. For this reason, we partition the domain into disjoint sampling regions, as we do in the particle-based scheme, and sample grid points and their associated raw histograms by region. The histogram values of any grid points that fall into the same sampling region are merged according to the normalization factor, N, and the net sampling volume, $\sum V_k$, where k is the grid point index. Note that the normalization factor is equal to the inverse of the total phase space volume where particles are sampled from and is computed during the simulation. The resulting histogram values are computed as follows:

$$H(i,j) = \frac{N}{\sum V_k} \sum h_k(i,j)$$

where $h_k(i,j)$ is the 2D histogram from a sampled grid point with index k and $H(i,j)$ is the resulting accumulated histogram.

3.3 Visualization System

Our system provides interactive control over the spatial organization of the histograms, the resolution of the histograms, and any velocity/frequency thresholding. While the user interacts with the

visualization, relevant information is automatically displayed according to the selection of the user. In addition, several features are included to extend the capabilities of the system and to aid in navigating the visualization. Lastly, the system provides basic control over 2D slice extraction from 3D datasets, as well as time step selection and animation.

3.3.1 The Histogram Layout

Different spatial histogram layouts can be selected by the user depending on the application. The spatial layout defines how the domain is partitioned into multiple local sampling regions. These local regions also define the location, size, and orientation of the visual representation of their associated histograms. Reference points are drawn at the centers of the histograms to mark the bin that corresponds to a velocity of zero.

Our primary layout uses a panning window approach. The sampling region is partitioned into a rectangular grid while a slider controls the number of partitions. When the user drags the mouse over the view space, or zooms in, the particles move relative to the screen, while the grid remains fixed. We also employ a layout that remains fixed with respect to the particles as the user pans and zooms. Such a layout has the benefit that zooming and panning can be done without changing the sampling region sizes. However, the panning window approach gives more control over the locations of the sampling regions with respect to the particles. In addition, custom layouts can be made to suit specific applications. An example of such a layout can later be seen in Figure 7 where the histograms are positioned and aligned along a magnetic flux surface.

3.3.2 Special Features

An example of the user interface can be seen in Figure 4 and highlights many of the features discussed in this section. As the user mouses over a histogram bin, corresponding particles are highlighted in green. This feature enables the histogram to double as a powerful selection tool for inspecting particle locations according to velocity and enables enhanced pattern and trend finding capabilities. Additionally, the histogram as a selection tool can be easily extended to trigger the display of other information about the corresponding particles, depending on the needs of the domain expert, e.g. temperature, trajectories, or velocity parallel to the viewing angle. Empty histogram cells are left transparent in order to minimize occlusion of the particles within the associated sampling regions. For the case when the entire histogram is filled with values, the opacity of the histograms can also be manually adjusted. In addition, the user can adjust the opacity of the particles themselves, to strike a balance between visibility and distraction.

The color vs. frequency mapping is, by default, globally assigned to all histograms in order for users to see the relative differences in frequencies between regions. This global thresholding, has the advantage of being able to reduce the influence of large peaks that would otherwise suppress subtle patterns in the histogram. The majority of the figures in this paper use the global thresholding scheme with the colormap shown in Figure 5. In addition, we also provide the option to locally scale the colormap normalizing with the maximum frequency in each histogram. This results in a separate color scale for each histogram which can be referenced by individually selecting the corresponding region. This has the advantage of clearly representing the distribution regardless of the number of particles that are sampled in that region. An example showing the difference between these schemes is later shown in Figure 8.

To help the user navigate the visualization, we provide a minimap that shows both the outline of the grid layout as well as the outline of the selected histogram. This feature gives users domain level context when exploring the data at high zoom levels. In addition, when the user selects a sampling region, the associated histogram is drawn larger at the side of the view space for easier viewing.

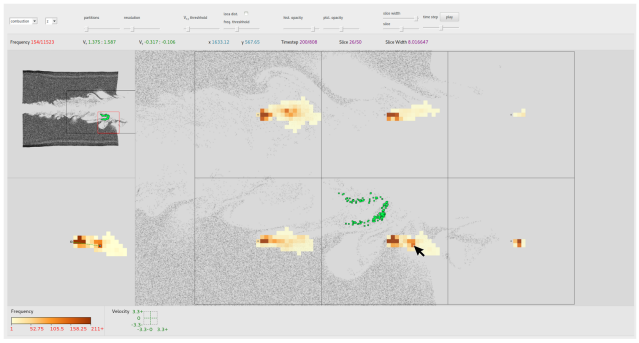

Figure 4: The user interface. The particles associated with the moused over histogram bin are shown in green. The moused over histogram is shown in more detail in the bottom left corner of the screen, and the mini-map is shown in the top left corner.

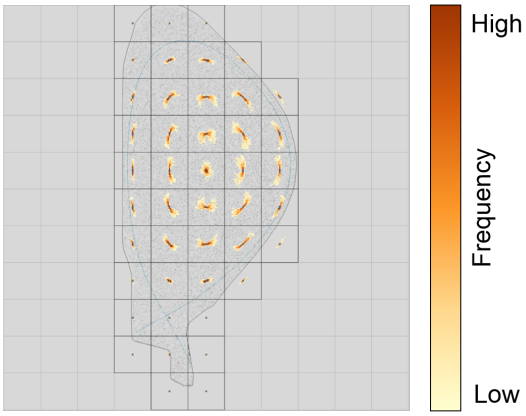

Figure 5: An overview of the histogram visualization using the XGC dataset showing the general patterns throughout the domain. The colormap used for the frequencies of each bin is shown as well and matches the colormap for all following figures.

This feature also allows the user to more accurately select specific histogram bins of interest when looking to identify additional information such as the exact value of the bin. In addition, a line is drawn from the center of the histogram to the selected bin to help the user perceive the direction of the associated velocity.

Other relevant information is automatically displayed as well. This includes information such as the velocity range corresponding to the selected histogram bin and the number of particles in the the sampling region. Additional domain specific information can also be displayed as needed. When applicable, controls are present for 2D slice extraction from 3D data sets. The user can control the slice thickness and position via sliders as well as the axis of the orthogonal view. A slider is also used to adjust the current timestep. Additionally, the system can cycle through timesteps automatically, freeing up the mouse to control other interactive parameters.

4 Results

We test our visualization tool with real world datasets to demonstrate its usefulness. First, we use a number of small and large-scale fusion datasets to test both the on-the-fly and in situ version of the workflow. The complex electromagnetic interactions between the positively charged ions and negatively charged electrons make these datasets difficult to study using traditional methods. We also use another flow-based dataset from a combustion simulation to demonstrate how our visualization tool handles 3D exploration as well.

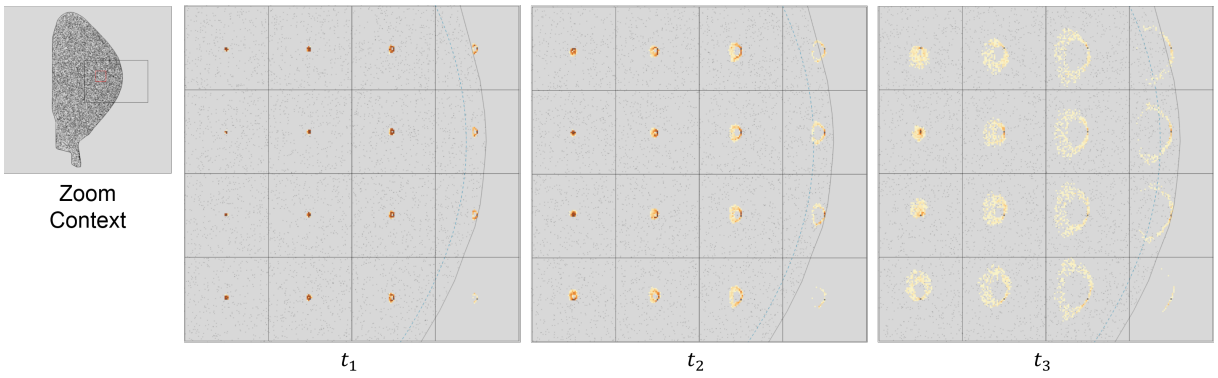

Figure 6: A zoomed view of the XGC dataset showing three select timesteps. Over time, unique ring-like patterns begin to form showing that slower moving particles tend to move in a vertical direction and faster moving particles tend to move towards the left. The presence of these rings seems to cycle on and off throughout the duration of the simulation.

Lastly, we test a geospatial dataset which tracks the movement of marine life in the Pacific Ocean to demonstrate its effectiveness in areas outside of flow visualization. In addition, we provide performance results to justify the interactivity of our technique using large datasets.

4.1 Fusion Datasets

Our fusion datasets are generated from large-scale simulation codes developed by scientists at Princeton Plasma Physics Laboratory. The first dataset comes from XGC [1], a simulation designed to study edge physics of magnetically confined plasmas. The particle dump from this simulation consists of $\sim 50,000$ particles with a Cartesian-based velocity.

The second dataset comes from GTS [16], a gyrokinetic simulation that focuses on studying microturbulence in fusion devices. We use this dataset to test the in situ generation scheme. This simulation contains ~ 500 million particles which were used to generate 3840 histograms on an unstructured grid, each with a resolution of 33 by 17 bins in situ. Instead of a Cartesian velocity decomposition, this dataset decomposes particle velocities according to "magnetic coordinates", a feature of interest to fusion scientists. In addition to the in situ generated histograms, ~ 15 million particles from the simulation were dumped for comparison and to test the on-the-fly generation scheme.

4.1.1 Particle-based Generation

The simulation domain of fusion devices represents a complex torus-like shape. However, scientists are often interested in motion of the plasma towards or away from chamber walls (in the direction of the "minor radius" of the torus). As a result, many visualizations (including our own) focus on presenting information on a 2D "poloidal" slice where the less interesting motion in the third dimension is hidden. We project all particles throughout the torus onto the slice view so that the resulting histogram-based visualization, while 2D in nature, represents information from the entire 3D domain.

Figure 5 shows an example of the histogram-based visualization for the XGC dataset. In this case, we directly sample any particles that fall into a histogram region (as denoted by the black grid). From this zoomed out overview image, we can see general patterns that permeate the domain. Since the center of each histogram/region represents a zero velocity, we can see that many particles, heavy ions in this case, are moving around the slice in both a clockwise and counterclockwise direction. While the majority are moving slowly (as shown in a dark color), smaller groups of particles are moving quickly (as shown in a lighter color). Particles near the top and bottom of the slice are either slow moving or stationary.

After obtaining an overview of the general patterns throughout the data, users can explore details by zooming into certain regions and adjusting the resolution of the histogram grid and the histograms themselves. Figure 6 shows a zoomed view of a select set of timesteps throughout the simulation. From the histograms we can tell that these particles, electrons in this case, begin by moving very slowly. Over time, their velocity begins to increase and ring-like patterns begin to form in the histograms. These rings show that the slower moving particles tend to move vertically while the faster moving particles tend to all move towards the left. Animating through the simulation shows that the presence of these ring-like structures cycles on and off at key timesteps.

Lastly, we demonstrate the ability to deform the histograms into non-Cartesian structures of interest. One such structure, the separatrix, represents an important magnetic flux surface in the tokamak geometry. In this case, we use a layout scheme which places histograms along the separatrix surface (defined in the acquired dataset), and transform the velocities into components parallel and perpendicular to the surface. This allows users to analyze the motion of particles relative to this curve. Figure 7 shows an overview of this new layout as well as a zoomed in version. Just like the Cartesian layout, the position of the histogram bin relative to the center of the histogram denotes the velocity direction that the bin represents. We can see that most of the particles tend to move parallel to the separatrix surface with the exception of some higher velocity particles which have a slight perpendicular component towards the center of the slice.

4.1.2 In Situ Generation

Next, we test the in situ scheme using the GTS simulation. In this case, a set of high resolution histograms have been pregenerated during the simulation using the full resolution particle data. These are then referenced using our sampling technique to generate an interactive visualization. Recall that the velocities from this simulation represent "magnetic coordinates" which have a complex orientation in 3D space. In these visualizations, the horizontal axis of the histogram represents the velocity of the particle parallel to the direction of the magnetic field, whereas the vertical axis represents the magnitude of the velocity perpendicular to the magnetic field. Since the perpendicular velocity is represented as a magnitude, it is always positive and results in only the top half of our histograms bearing values. These types of velocity plots are commonly used when studying the results of the GTS simulation.

Figure 8 shows an example of the in situ sampled histograms using our visualization tool (left images) vs. a comparison to the

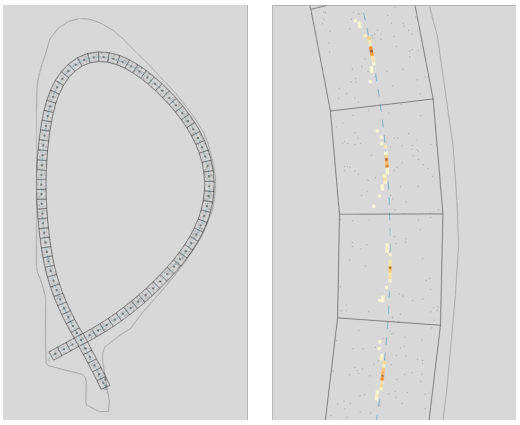

Figure 7: Left) An overview showing the histogram layout on the separatrix surface. Right) A zoomed view showing that only higher velocity particles have a perpendicular velocity component to this surface.

on-the-fly particle-based sampling scheme using a smaller subset ($\sim 3\%$) of particle data dumped from the simulation (right images). The top images use the global thresholding scheme to show realtive differences between histograms while the bottom images use the local color-mapping scheme to clearly show all distribution functions. The points in blue show the grid points over which histograms were sampled. From the two images it is clear that the in situ version was able to capture more detail in the velocity decompositions since it represents statistical information from a much larger sample size.

4.2 Combustion Dataset

The combustion dataset we use comes from S3D, a peta-scale combustion simulation developed by scientists at Sandia National Labs [18]. This particular simulation run depicts a 3D highly-turbulent lifted ethylene jet flame which can lead to a better understanding of phenomena such as autoignition in engines. In this dataset, being able to explore all three dimensions of the domain is important to combustion scientists. Since our histogram-based visualization is 2D in nature, we use the interactivity of the system to explore different aspects of the domain. For example, users can choose to view the data from multiple orthogonal axes. In addition, the histograms can represent particles sampled from a slice in the data whose position and thickness are both adjustable in the user interface.

From the histogram-based view in Figure 9, we can see the velocity decomposition of the particles from multiple orthogonal viewing angles. Viewing from the front, one can see that particles are moving towards the right both quickly and slowly near the center of the jet, whereas particles near the top and bottom of the image are stationary. In addition, mousing over a particular histogram bin highlights the corresponding set of particles in green. Viewing from the side shows similar patterns but reveals an additional dimension of motion. Upon careful inspection, there is a greater variation in the velocities in the top half of the image when compared to the bottom half.

4.3 Marine Life Movement Dataset

The last data we test comes from the Tagging of Pelagic Predators (TOPP) [2] dataset. This is a geospatial movement dataset which tracks the motion of marine life throughout the Pacific Ocean. Datasets like these are essential in understanding animal behavior in terms of migratory or feeding patterns. We choose this dataset to show that our histogram-based visualization tool has applicability in areas outside of flow visualization.

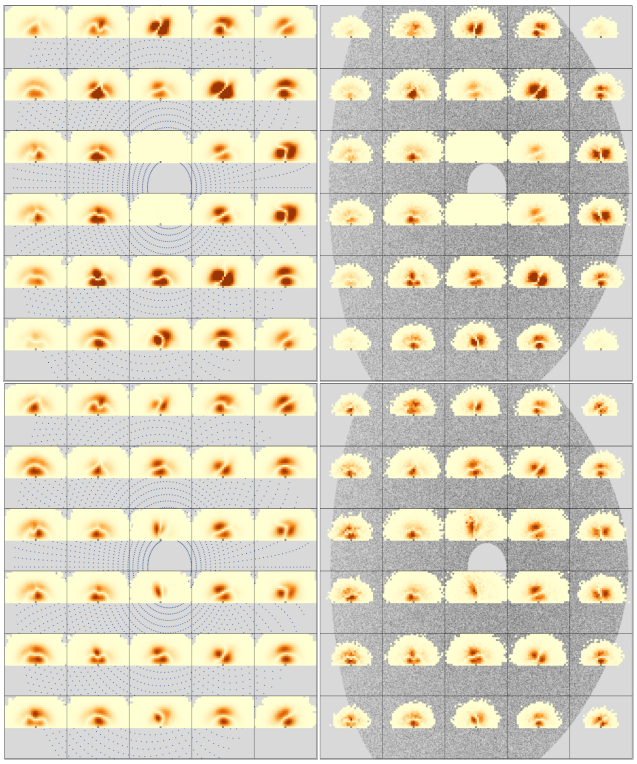

Figure 8: Left-column) Visualizing the in situ generated histograms. Right-column) Visualizing a smaller subset of the particle data from the simulation using the on-the-fly generation method. Top-row) Using global thresholding to color bins. Bottom-row) Using the local color-mapping to color bins.

Figure 10 shows an example of our visualization applied to this geospatial movement dataset. More specifically, we look at the migratory patterns of elephant seals which inhabit the California coast. On the left we can see that the primary direction of motion is towards the north-west in late February and early March. This is a result of the seals migrating into the Pacific Ocean to feed after a long breeding period on the coast. A few months later in late April and early March, the direction of motion reverses towards the south-east when many of the seals return to the coast.

4.4 Performance Results

We provide a set of performance results for both the on-the-fly particle-based generation, as well as the in situ generation scheme to demonstrate the interactive capabilities of the visualization tool. We use an Intel i5 processor with an Nvidia GTX 970 graphics card to conduct these tests. In the particle-based generation scheme, the step that constructs the histograms by assigning all particles to a histogram bin and accumulating the results is the performance bottleneck. Using the GPU accelerated method described earlier, we are able to generate histograms from large sets of particles at an interactive pace. Figure 11 shows timing results for the on-the-fly histogram generation as a function of the number of particles that need to be sampled. After reaching the maximum sizes of our available datasets, we artificially inject additional particles to test larger scales. The blue curve represents the total time to generate the visualization. The red curve represents the histogram generation time on the GPU as well as the CPU/GPU transfer time, and assumes that the dataset has already been loaded into memory. We can see that over 100 million particles can be transformed into the histogram-based representation in less that 100 ms, and that this tends to scale

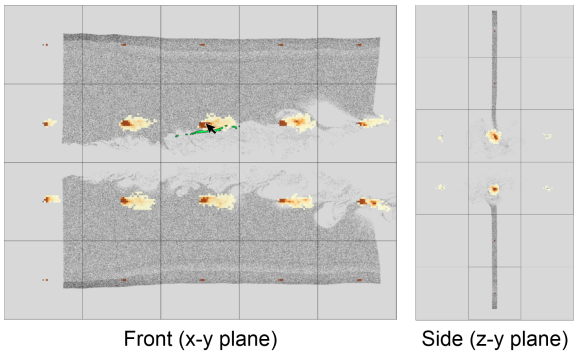

Figure 9: Two images showing the combustion dataset as viewed from different orthogonal directions. From the x-y plane it is clear that particles near the center of the combustion jet are moving both quickly and slowly to the right. Particles near the top and bottom of the images are fairly stationary. A set of particles is highlighted in green when mousing over a particular histogram bin of interest.

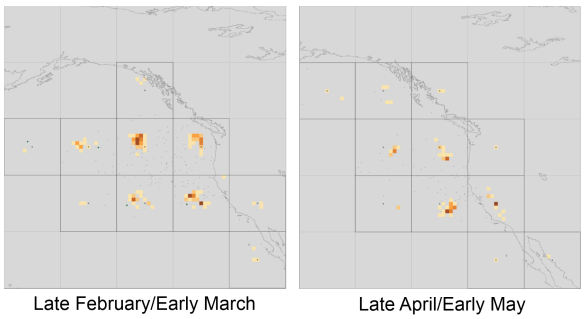

Figure 10: The velocity decomposition of elephant seals at two different times of the year near the California coast. Left) The direction of movement is primarily towards the north-west as the seals migrate into the Pacific Ocean to feed. Right) The direction of movement is primarily to the south-east as the seals return to the coast.

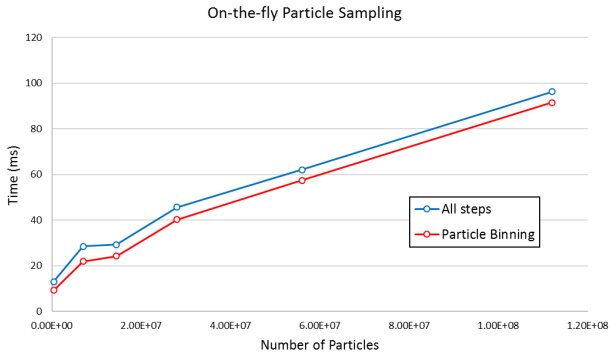

Figure 11: A graph showing the computation time for the on-the-fly histogram generation vs. particle sampling. We can see that over 100 million particles can be visualized at interactive speeds.

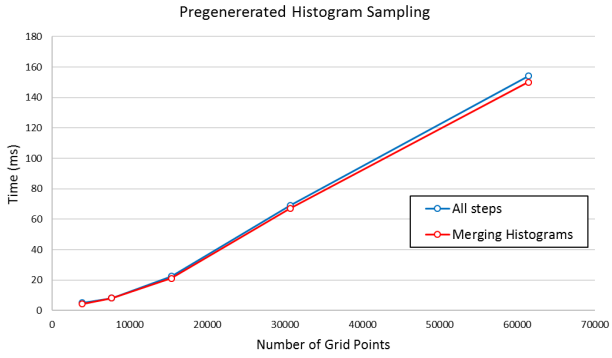

Figure 12: A graph showing the computation time for the sampling of high resolution histograms generated in situ. We can see that over 70,000 histograms can be interactively merged with very little computational overhead.

linearly. Note that since each particle must be mapped to a histogram bin regardless of the number of possible bins, adjusting the number of histograms and their resolution has an insignificant impact on the overall computation time.

When there is a need to visualize the statistical information for extreme-scale applications, we switch to the in situ sampling scheme. The allocation of computing resources to pregenerate the histograms is dependent on the type of simulation and domain specific application. Using the GTS fusion simulation as an example, the cost to generate the high resolution histograms and save them to disk is not insignificant. However, this is a calculation that does not need to be completed for every timestep. As a result, histograms can be saved at only key points throughout the simulation making their overall computational cost small in comparison.

Once the in situ generation is complete, the bottleneck in the visualization tool is once again the time to sample and transform the high resolution histograms into the version presented to the user (according to view/zoom and other interactive settings). Figure 12 shows the timing results as a function of the number of grid points that need to be merged. The blue curve represents the total time to generate the visualization. The red curve represents the total time it takes to sample and merge the high resolution histograms from grid points into the representation demanded by interactive settings. Once again, after reaching the maximum sizes of our available datasets, we artificially include additional grid points to test larger scales. We can see that our sampling scheme can interactively generate the visualization result from over 70,000 grid points and that this tends to scale linearly.

5 Discussion

The above results demonstrate the usefulness of our visualization system to study the velocity decomposition of a group of objects. Not only can this system be used to study various applications, it can also be used in both large and small-scale settings.

5.1 Scalability

As data becomes large, there are several steps we employ to maintain interactivity. A mid-level desktop computer can handle the system with around 100 million raw samples before there is a hindering delay when changing interactive settings. We are able to exploit GPU acceleration for the computationally intensive procedures to the point where memory limitations of the hardware are likely to become the next bottleneck. If the data is too large for a single desktop, this scheme could easily be transformed to a distributed setting. Each spatial partition of our histogram layout can be computed independently with little communication overhead.

In extreme-scale applications, we utilize a set of high resolution histograms generated during the simulation to capture the statistical information of the full particle representation. The main advantage of this technique is that this resulting data representation is much

more manageable and statistical information from the full simulation data is retained. As long as the histograms are generated in situ at a high enough resolution, our visualization system can reproduce an interactive exploration with negligible perceptual loss.

The next step involves direct integration into the simulations themselves. This will allow researchers to visualize results as simulations are run, without the overhead from expensive I/O costs. In addition, scientists could choose to steer the simulation based on the visualization.

5.2 Applicable Data Types

In this paper, we demonstrate the applicability of our visualization system to particle-based flow data as well as geospatial movement data. However, there are other forms of data which can benefit from this system as well. For example, while the histograms used in our visualization system are designed to represent a velocity decomposition, they can instead be used to represent other parameters and variables present in the data. Another application could be for use in studying multi-dimensional data. Our 2D histograms embedded in a 2D screen space layout allows users to visualize information from four different parameters at once (two position parameters and two velocity parameters). This system could be adapted to effectively study trends between many variables simultaneously.

5.3 Interpolation of Pregenerated Histograms

Another future improvement of the system is a more sophisticated sampling scheme of the pregenerated histograms. Currently, the system samples the entire grid point's distribution function when if falls into a specific sampling region. However, given certain view/pan settings, it is possible that some grid points lie on the boundary between two regions that we sample from. Since these grid points represent volumes, part of the volume can fall into a neighboring sampling area. Saving the geometric shape of each grid point volume from the simulation as well can allow the visualization system to intelligently weight points according to the fraction of its volume that is being sampled. While this will improve the accuracy of the visualization result, it is still unclear whether it will be perceptually noticeable to users.

6 Conclusion

Overall, we present an interactive system which utilizes the advantages of velocity-based histograms to visualize and explore the motion of a group of objects. While many traditional techniques are subject to a trade-off between visual clutter and loss of detail, a histogram-based representation can provide a clear description of the velocity decomposition of a system of study. We employ two schemes: an on-the-fly histogram generation technique which directly samples particle information from small and medium-scale datasets, and an in situ scheme for extreme-scale applications which samples high resolution histograms generated during simulation time. We demonstrate the effectiveness of our visualization system using a number of fundamentally different real world datasets and justify its interactive capabilities through performance evaluations.

Acknowledgements

This research is sponsored in part by the U.S. National Science Foundation via grants NSF DRL-1323214 and NSF IIS-1320229, and also by the U.S. Department of Energy through grants DE-SC0005334, DE-FC02-12ER26072, and DE-SC0012610. We would also like to thank Princeton Plasma Physics Laboratory for the fusion datasets, Sandia National Laboratory for the combustion dataset, and the TOPP collaborators for the marine life dataset.

References

[1] M. Adams, S.-H. Ku, P. Worley, E. D'Azevedo, J. Cummings, and C.-S. Chang. Scaling to 150k cores: Recent algorithm and performance engineering developments enabling xgc1 to run at scale. *Journal of Physics: Conference Series.*, 180(1), 2009.

[2] B. A. Block, I. D. Jonsen, S. J. Jorgensen, A. J. Winship, S. A. Shaffer, S. J. Bograd, E. L. Hazen, D. G. Foley, G. A. Breed, A.-L. Harrison, J. E. Ganong, A. Swithenbank, M. Castleton, H. Dewar, B. R. Mate, G. L. Shillinger, K. M. Schaefer, S. R. Benson, M. J. Weise, R. W. Henry, and D. P. Costa. Tracking apex marine predator movements in a dynamic ocean. *Nature*, 475:86–90, July 2011.

[3] N. Dalal, B. Triggs, and C. Schmidi. Human detection using oriented histograms of flow and appearance. *Computer Vision - ECCV 2006*, 3952:428–441, dec 2006.

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

[5] M. Hlawatsch, P. Leube, W. Nowak, and D. Weiskopf. Flow radar glyphs - static visualization of unsteady flow with uncertainty. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):1949–1958, Dec 2011.

[6] N. Ihaddadene and C. Djeraba. Real-time crowd motion analysis. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, Dec 2008.

[7] C. Jung, L. Hennemann, and S. Raupp Musse. Event detection using trajectory clustering and 4-d histograms. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1565–1575, Nov 2008.

[8] R. M. Kirby, H. Marmanis, and D. H. Laidlaw. Visualizing multivalued data from 2d incompressible flows using concepts from painting. In *Proceedings of the Conference on Visualization '99: Celebrating Ten Years*, VIS '99, pages 333–340, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.

[9] E. Marsch, X.-Z. Ao, and C.-Y. Tu. On the temperature anisotropy of the core part of the proton velocity distribution function in the solar wind. *Journal of Geophysical Research*, 109:A04102, 2004.

[10] H. Obermaier and K. Joy. Derived metric tensors for flow surface visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2149–2158, Dec 2012.

[11] M. S. Paoletti, M. E. Fisher, K. R. Sreenivasan, and D. P. Lathrop. Velocity statistics distinguish quantum turbulence from classical turbulence. *Physical Review Letters*, 101:154501, Oct. 2008.

[12] A. Romanoni, M. Matteucci, and D. G. Sorrenti. Background subtraction by combining temporal and spatio-temporal histograms in the presence of camera movement. *Machine Vision and Applications*, 25(6):1573–1584, dec 2014.

[13] M. Salewski, B. Geiger, A. Jacobsen, M. Garca-Muoz, W. Heidbrink, S. Korsholm, F. Leipold, J. Madsen, D. Moseev, S. Nielsen, J. Rasmussen, M. Stejner, G. Tardini, M. Weiland, and the ASDEX Upgrade Team. Measurement of a 2d fast-ion velocity distribution function by tomographic inversion of fast-ion d-alpha spectra. *Nuclear fusion*, 54(2):023005, Feb. 2014.

[14] D. Thompson, J. Levine, J. Bennett, P.-T. Bremer, A. Gyulassy, V. Pascucci, and P. Pebay. Analysis of large-scale scalar data using hixels. In *Large Data Analysis and Visualization (LDAV), 2011 IEEE Symposium on*, pages 23–30, Oct 2011.

[15] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann. A Survey on Interactive Lenses in Visualization. In R. Borgo, R. Maciejewski, and I. Viola, editors, *EuroVis - STARs*. The Eurographics Association, 2014.

[16] W. X. Wang, Z. Lin, W. M. Tang, W. W. Lee, S. Ethier, J. L. V. Lewandowski, G. Rewoldt, T. S. Hahm, and J. Manickam. Gyrokinetic simulation of global turbulent transport properties in tokamak experiments. *Phys. Plasmas*, 13(19):092505, Oct. 2006.

[17] C. Wittenbrink, A. Pang, and S. Lodha. Glyphs for visualizing uncertainty in vector fields. *Visualization and Computer Graphics, IEEE Transactions on*, 2(3):266–279, Sep 1996.

[18] C. S. Yoo, E. Richardson, R. Sankaran, and J. Chen. A dns study on the stabilization mechanism of a turbulent lifted ethylene jet flame in highly-heated coflow. *Proceedings of the Combustion Institute*, 33(1):1619–1627, Oct. 2011.