

Winding Angle Assisted Particle Tracing in Distribution-Based Vector Field

Cheng Li
The Ohio State University
li.4076@osu.edu

Han-Wei Shen
The Ohio State University
shen.94@osu.edu

ABSTRACT

Distribution models are widely used for data reduction applications. The Gaussian mixture model (GMM) is a powerful tool to capture multiple-peak distributions. For distribution-based vector field datasets represented by GMM, there are still loss of information which sometimes causes too much error when performing flow line tracing tasks. As a compensation, we analyze the vector transition pattern between consecutive vector directions. The vector transition is depicted by distributions of winding angles. When performing streamline and pathline tracing, we utilize the winding angle to estimate a conditional distribution of local vectors, using the Bayes Theorem. The conditional distribution can be used for both Monte Carlo flow line tracing, and single flow line tracing. We applied our distribution model on data reduction applications, and demonstrated that improved flow line tracing quality was achieved.

CCS CONCEPTS

•Computing methodologies →Parametric curve and surface models;

KEYWORDS

Distribution, Data Reduction, Flow Field, Particle Tracing

ACM Reference format:

Cheng Li and Han-Wei Shen. 2017. Winding Angle Assisted Particle Tracing in Distribution-Based Vector Field. In *Proceedings of SA '17 Symposium on Visualization, Bangkok, Thailand, November 27-30, 2017*, 8 pages. DOI: 10.1145/3139295.3139297

1 INTRODUCTION

Distribution models are frequently used to measure and analyze uncertainty in scientific datasets. It can also be used to preserve the salient statistical information in the data, and to recover as much lost information as possible through statistical inference and sampling. Many distribution models, such as the Gaussian distribution, the Gaussian mixture model, and histograms have been applied to represent datasets for various visualization tasks. When these techniques are applied to block-based data, that is, replacing raw data in each data chunk by distributions, thanks to their compactness, substantial data reduction can be achieved.

Flow line tracing is a widely used technique in vector field visualization. However, distribution models for vector data contain

ambiguity due to the loss of information, which causes difficulties when determining the direction of particle advection. As a consequence, flow lines traced in the reduced vector field can easily drift away from the flow lines in the original dataset. Using the Monte Carlo sampling method to form a bundle of sample traces is a popular strategy to represent the range of possible flow lines. But the trace bundle itself can bear larger errors when there exist complex flow behaviors. Therefore, algorithms for more reliable streamline and pathline tracing in the distribution-based vector field are highly desired.

In this paper, we present a new flow line tracing algorithm for 2D distribution-based vector fields. The main idea is to exploit the information obtained from adjacent vectors, called the vector transition information, as a supplement for the vector data distribution. We use blocks to subdivide a vector field dataset, and model the vector data within each block by Gaussian mixture model (GMM). At the same time, our system also analyzes the transition of consecutive integration directions at each grid point. This is done by processing a hypothetical particle at the grid point, and computing the winding angle. The distributions of winding angles are also modeled as Gaussian mixtures. The modeled vector transition information is used when tracing streamlines and pathlines in the reduced flow field. The vector GMM stored in the reduced field is essentially an unconditional distribution of local vectors. When tracing a flow line, we treat the integration direction used in the last step as a precondition of the current step. By checking how well the directions in these two consecutive steps fit the pre-stored vector transition information, we derive a conditional distribution of local vectors. This conditional vector distribution is used to retrieve a direction for integration. We apply the conditional vector distribution for two tracing tasks. The first task is to use the Monte Carlo sampling technique to generate a trace bundle. By placing a large number of Monte Carlo seeds, we integrate them using directions randomly sampled from the conditional distribution. The second task is to trace a single flow line in a more reliable way. We use the mean direction of the conditional distribution to integrate the particle. Due to extra information from the winding angle distribution, our methods can achieve improved quality than the baseline methods for both tasks.

2 RELATED WORK

2.1 Distribution-Based Visualization

Distribution-based visualization techniques play important roles in uncertain data analysis. They have been commonly applied to study ensemble datasets, or datasets reduced in a block-wise manner. Liu *et al.* [Liu et al. 2012] used the Gaussian mixture model to represent the data distribution, and performed volume rendering by Monte Carlo sampling. Chaudhuri *et al.* [Chaudhuri et al. 2012]

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SA '17 Symposium on Visualization, Bangkok, Thailand

© 2017 Copyright held by the owner/author(s). 978-1-4503-5411-0/17/11...\$15.00
DOI: 10.1145/3139295.3139297

applied histograms to describe the blocks of large datasets, and conducted information theory based visualization tasks. Thompson *et al.* [Thompson *et al.* 2011] enriched the histogram to *hixel*, to perform topological feature extraction and uncertain iso-surface rendering. Isocontours in uncertain scalar fields were also studied using various of distribution models [Athawale and Entezari 2013; Athawale *et al.* 2016; Pfaffelmoser *et al.* 2011; Pöthkow and Hege 2011; Pöthkow *et al.* 2011]. Dutta *et al.* modeled data in blocks by Gaussian mixture model, and performed feature detection [Dutta and Shen 2016] and anomaly detection [Dutta *et al.* 2017].

2.2 Visualization of Flow Uncertainty

Besides scalar fields, we summarize distribution-based studies conducted on uncertain vector fields in this subsection. *Texture-based rendering*: Botchen *et al.* [Botchen *et al.* 2005] used texture-based techniques to visualize the uncertainty in 2D flow fields. *Flow critical points detection*: Otto *et al.* [Otto *et al.* 2010, 2011] and Petz *et al.* [Petz *et al.* 2012] modeled uncertain vector data by Gaussian distributions to detect the probability distribution of critical points. *Single flow line with error estimation*: Love *et al.* [Love *et al.* 2005] rendered the possible range of each node of the flow line, by placing a circle on this node. The National Hurricane Center [Center 2016] visualized the estimated error of the prediction of a hurricane pathline by a cone shape. Cox *et al.* [Cox *et al.* 2013] and Liu *et al.* [Liu *et al.* 2015] used Monte Carlo sampling to visualize the uncertainty of possible hurricane paths or hurricane positions. *Bundle of flow lines*: When a bundle of possible traces for a particle is at hand, Bürger *et al.* [Bürger *et al.* 2012] measured its uncertainty by creating a visitation map of the 3D space, and showed the result by volume rendering; Mirzargar *et al.* [Mirzargar *et al.* 2014] extended the concept of scalar boxplot to the curve boxplot; and Ferstl *et al.* [Ferstl *et al.* 2016] did clustered the flow lines in a lower dimension space, and visualized the clusters a shape of lobes. *Comparing each ensemble member from an ensemble set*: Guo *et al.* [Guo *et al.* 2013] measured the variation by computing the distances of the Lagrangian specifications; Jarema *et al.* [Jarema *et al.* 2015, 2016] measured similarity by analyzing the pathlines in each member; and Ribeiro *et al.* [Ribeiro *et al.* 2016] used the Helmholtz-Hodge decomposition for 2D vector fields. *Fiber tracking in diffusion MRI imaging*: this field is related because the voxel of a diffusion MRI image is given by a discrete distribution on directions, and fibers can be generated using similar integration methods as flow lines. Researchers used Bayes Theorem [Friman *et al.* 2006; Zhang *et al.* 2009] or the Kalman filter [Malcolm *et al.* 2010] to estimate the integration of brain fibers, with prior information from the previous step being considered to guide the integration.

3 DATA REDUCTION MODEL

3.1 Model Vector Data Distribution by GMM

Given a 2D vector field dataset, we partition the data space into non-overlapping blocks covering the space. In each block, the vectors $\{\vec{a}_0, \dots, \vec{a}_i, \dots, \vec{a}_{N-1}\}$ are modeled into a multivariate distribution using the Gaussian mixture model (GMM), denoted by:

$$g(\vec{v}) = \sum_{j=1}^K w_j \mathcal{N}(\vec{v} | \mu_j, \Sigma_j), \quad (1)$$

where K is the number of Gaussian components adopted for the model. Each component is characterized by the mean vector μ_j and the covariance matrix Σ_j , and is associated with a weight w_j where all weights sum up to 1. As a frequently used model for block-wise data reduction [Dutta *et al.* 2017; Dutta and Shen 2016; Liu *et al.* 2012; Wang *et al.* 2017], GMM can capture data statistical properties when the distribution follows a multiple-peak shape, for which a single Gaussian distribution does not work effectively. Figure 1 gives an overview of our partition and reduction procedure. Figure 1(b) shows a block which contains two distinctive vector patterns, one pointing upper-right, the other one pointing bottom-right. Figure 1(c) shows modeling these vectors into a mixture of two Gaussian components.

The estimation of the GMM is to estimate the parameters $\{w_j, \mu_j, \Sigma_j\}$, and is conducted using the Expectation-Maximization (EM) method [Bilmes *et al.* 1998]. We skip method details here, but only introduce the probability model P_i used during the procedure. The EM method works iteratively. At each step, it not only updates the parameters $\{w_j, \mu_j, \Sigma_j\}$, but also updates a hidden discrete probability model P_i for each data sample \vec{a}_i . For any \vec{a}_i , $P_i(j)$ means the probability that this sample comes from the j th Gaussian component, and $\sum_{j=1}^K P_i(j) = 1$. In the end of the iterative loop, $P_i(j)$ is finalized, and the model parameters $\{w_j, \mu_j, \Sigma_j\}$ are achieved using the following equations:

$$\begin{aligned} w_j &= \sum_{i=0}^{N-1} P_i(j) / N \\ \mu_j &= \sum_{i=0}^{N-1} P_i(j) \vec{a}_i / \sum_{i=0}^{N-1} P_i(j) \\ \Sigma_j &= \sum_{i=0}^{N-1} P_i(j) (\vec{a}_i - \mu_j)(\vec{a}_i - \mu_j)^T / \sum_{i=0}^{N-1} P_i(j) \end{aligned} \quad (2)$$

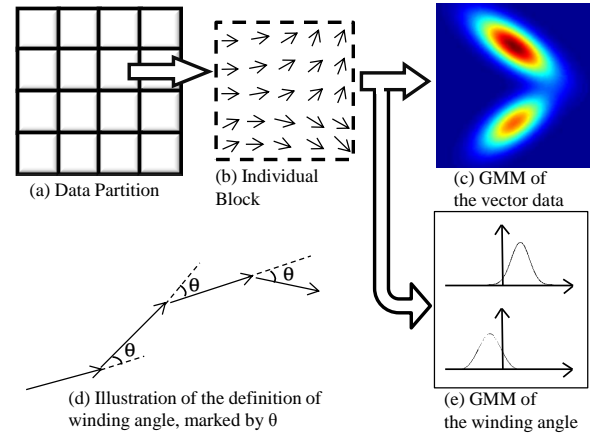


Figure 1: Overview of our reduction model. Data in one block as (b) is modeled into vector GMM as in (c), plus winding angle GMM as (e).

3.2 Model the Vector Transition

Despite various benefits of using distributions for data representation, there is inevitable information loss in the process. For GMM, while this model can perfectly describe important statistical features of the data, its power may fail to extend to the sub-block level.

Use Figure 1 as an example. The GMM model in Figure 1(c) uses two Gaussian components to describe two vector patterns in Figure 1(b). However this GMM model does not contain the information that one Gaussian component mainly represents the vector pattern at the top three rows of the block in Figure 1(b), while the other Gaussian component represents the bottom two rows. In this case, what has been lost is spatial information. Directly modeling the spatial information of distributions has been studied [Wang et al. 2017], for visual analytical tasks that require deterministic spatial locations such as isosurface or volume rendering. However, when performing flow line tracing, the location of the particle at each step is highly impacted by previous tracing status, thus is not deterministic but with large uncertainty, especially for distribution-based data. For example, when a particle travels into the block in Figure 1(b), it may not be proper to grant more preference to the first Gaussian component just because the particle is located at the top half, or prefer the second Gaussian component if at the bottom.

To compensate for the information loss, we use distributions of vector transition to fulfill this goal. The vector transition information tells us that during the integration of a flow line, how the vector direction of the last step influences the direction of the current step. With this information, the last integration direction of the particle can serve as a supplemental information to give a more robust estimate of the current step. Using the relation between consecutive directions of the tracing has been adopted in several studies of brain fiber tracking on diffusion MRI data. Previous researchers assumed that brain fibers are generally smooth with a slight change of the advection direction at a time. Based on this assumption, they described the probability distribution of a predicted direction, which will have a higher probability when it has a smaller angle difference from the direction in the previous step [Friman et al. 2006; Zhang et al. 2009]; or directly used the same direction with the last step as a direction prediction [Malcolm et al. 2010]. Extended from these early inspiring methods, we try to quantify this direction change, instead of just assume it to be a slight one. Therefore, our method can also handle turbulent flow where turning between consecutive directions is large, or compound regions where multiple vector transition patterns exist.

To model the vector transition pattern, we measure it using the *winding angle* [Shi et al. 2009], which evaluates the angle of consecutive directions along a flow line, as illustrated in Figure 1(d). At each block, with the input data $\{\vec{a}_0, \dots, \vec{a}_{N-1}\}$, we compute the winding angle θ_i for each sample \vec{a}_i . This is conducted by placing a hypothetical particle at the position p_i of \vec{a}_i , and carrying out a backward integration with a selected step length L . Then the particle is driven to the position $p'_i = p_i - L \times \vec{a}_i$. At the position p'_i of the L earlier time frame, we retrieve the vector data \vec{a}'_i . At last θ_i is computed as the rotation angle from \vec{a}'_i to \vec{a}_i . Therefore in this block, we achieve a set of winding angles $\{\theta_0, \dots, \theta_{N-1}\}$.

These angles $\{\theta_0, \dots, \theta_{N-1}\}$ will also be modeled using GMM denoted by weights w_j^θ , mean μ_j^θ and variance Σ_j^θ . However instead of estimating an independent GMM, we closely associate it with the GMM used to model the vector data. The reason is that after applying the EM method for the vector data, we have achieved a set of probability $P_i(j)$ used in Equation 2. This is essentially a probabilistic classifier, which assigns each sample element to

the cluster j by a probability. Because θ_i and \vec{a}_i are essentially both attributes of the same sample element, they should have the same probability of belonging to each cluster. Thus for the angle GMM, the weights for its components w_j^θ become the same with the weights w_j in Equation 2, and the mean and variance of each Gaussian component are directly computed by:

$$\begin{aligned} \mu_j^\theta &= \sum_{i=0}^{N-1} P_i(j) \theta_i / \sum_{i=0}^{N-1} P_i(j) \\ \Sigma_j^\theta &= \sum_{i=0}^{N-1} P_i(j) (\theta_i - \mu_j^\theta) (\theta_i - \mu_j^\theta)^T / \sum_{i=0}^{N-1} P_i(j), \end{aligned} \quad (3)$$

without the necessity to run another EM loop again. Finally the distribution can be written as:

$$h(\theta) = \sum_{j=1}^K w_j \mathcal{N}(\theta | \mu_j^\theta, \Sigma_j^\theta) \quad (4)$$

As a result, within the same block, the j th component of the angle GMM actually only depicts the transition pattern of the vectors in the sample space described by the j th component of the vector GMM. Figure 1(e) shows the winding angle distribution for the block in Figure 1(b).

4 TRACING METHOD

In a distribution-based flow field, tracing flow lines is usually done by exploiting the local distribution of vectors $g(\vec{v})$. $g(\vec{v})$ represents the probability density of getting a sample vector of certain value at this location. This is an unconditional distribution, meaning it best describes the probability density when there exists no prior condition that may affect this sampling. However, during flow line tracing, we are under the condition that we have already traced the particle for several steps. This condition actually changes the sample space of the local vector, and a conditional vector distribution will better describe the sampling space.

Assume the flow line tracing is a Markov model, which means that the impact to the current state made by the whole history is equivalent to the impact made by the previous event. Therefore, we only need to consider the influence made by the integration direction from the previous step. Denote the direction at the k th step as \vec{v}_k , whose unconditional distribution is $g(\vec{v})$ as in Equation 1. During an actual tracing process, when the last integration direction \vec{v}_{k-1} is known, we can estimate a conditional distribution $g(\vec{v} | \vec{v}_{k-1})$, to better describe the sampling space of \vec{v}_k .

Using the Bayes Theorem, we have the conditional distribution of the vector data as:

$$g(\vec{v} | \vec{v}_{k-1}) = g(\vec{v}_{k-1} | \vec{v}) \times g(\vec{v}) / g(\vec{v}_{k-1}) \quad (5)$$

$g(\vec{v}_{k-1})$ can be neglected since it acts as a normalization coefficient. The estimation of $g(\vec{v}_{k-1} | \vec{v})$ will make use of the distribution of the winding angle we have stored in our reduction model. In our method, we assume the impact from the length of \vec{v}_{k-1} can be ignored, so $g(\vec{v}_{k-1} | \vec{v})$ can be transformed into the conditional probability of the winding angle formed by \vec{v}_{k-1} and \vec{v} . Use $\Theta(\vec{v}_{k-1}, \vec{v})$ to denote this angle, therefore we have

$$\begin{aligned} g(\vec{v} | \vec{v}_{k-1}) &= \alpha \times g(\vec{v}) \times g(\vec{v}_{k-1} | \vec{v}) \\ &= \alpha \times g(\vec{v}) \times P(\Theta(\vec{v}_{k-1}, \vec{v}) | \vec{v}) \end{aligned} \quad (6)$$

Here α is a normalization coefficient to make the distribution integral to one. We already have an unconditional distribution of winding angles as in Equation 4. $\Theta(\vec{v}_{k-1}, \vec{v})$ will follow this unconditional distribution after scaled by L/l , where l is the step length that \vec{v}_{k-1} has traveled, and L is the step length used to model the winding angle distribution (as in Paragraph 3 of Section 3.2). This scaling is to compensate for the different degrees of vector transition caused by different step lengths. Denote the scaled angle by $\theta(\vec{v}_{k-1}, \vec{v})$, i.e., $\theta(\vec{v}_{k-1}, \vec{v}) = \Theta(\vec{v}_{k-1}, \vec{v}) \times L/l$, then we have $\theta(\vec{v}_{k-1}, \vec{v})$ following the $h(\theta)$ in Equation 4. The condition of this probability \vec{v} does not directly affect the Gaussian components of $h(\theta)$, but can affect the weights w_j . It is because after knowing the value of \vec{v} , its probability of coming from the j th component is not simply w_j , but a conditional distribution $\pi(j|\vec{v})$; this makes its winding angle also have a probability $\pi(j|\vec{v})$ of coming from the j th component of $h(\theta)$. So we continue the deduction as:

$$\begin{aligned}
 g(\vec{v}|\vec{v}_{k-1}) &= \alpha \times g(\vec{v}) \times h(\theta(\vec{v}_{k-1}, \vec{v})|\vec{v}) \\
 &= \alpha g(\vec{v}) \sum_{j=1}^K \pi(j|\vec{v}) \times \mathcal{N}(\theta(\vec{v}_{k-1}, \vec{v})|\mu_j^\theta, \Sigma_j^\theta) \\
 &= \alpha g(\vec{v}) \sum_{j=1}^K \frac{w_j \mathcal{N}(\vec{v}|\mu_j, \Sigma_j)}{g(\vec{v})} \times \mathcal{N}(\theta(\vec{v}_{k-1}, \vec{v})|\mu_j^\theta, \Sigma_j^\theta) \\
 &= \alpha \sum_{j=1}^K (w_j \mathcal{N}(\theta(\vec{v}_{k-1}, \vec{v})|\mu_j^\theta, \Sigma_j^\theta)) \times \mathcal{N}(\vec{v}|\mu_j, \Sigma_j)
 \end{aligned} \tag{7}$$

Directly applying the current result is hard, since it has an angle computation nested in an inner product of two Gaussian mixtures. So we make a simplification by replacing the angle of \vec{v}_{k-1} and \vec{v} , to the angle of \vec{v}_{k-1} and the mean vector of the corresponding Gaussian component, i.e., μ_j . Therefore we have:

$$g(\vec{v}|\vec{v}_{k-1}) = \alpha \sum_{j=1}^K (w_j \mathcal{N}(\theta(\vec{v}_{k-1}, \mu_j)|\mu_j^\theta, \Sigma_j^\theta)) \times \mathcal{N}(\vec{v}|\mu_j, \Sigma_j) \tag{8}$$

Given \vec{v}_{k-1} and j , the angle $\theta(\vec{v}_{k-1}, \mu_j)$ can be computed into a number. Therefore, the conditional distribution $g(\vec{v}|\vec{v}_{k-1})$ becomes a new GMM distribution.

We can observe that our conditional GMM has the same Gaussian components with the unconditional distribution, but with different weights. An intuitive interpretation of this phenomenon is, given the previous integration direction, if a Gaussian component fits its corresponding transition pattern better, it will be rewarded by an increased weight.

Now with the conditional vector distribution at hand, in the following two subsections, we discuss how to use it to perform two tracing tasks. The first task is to generate a trace bundle by using a large number of Monte Carlo seeds. The second task is to trace single flow line in the distribution field.

4.1 Monte Carlo Tracing

Monte Carlo sampling is widely used to study an uncertain particle route, by placing a large number of Monte Carlo seeds at the initial location to perform the tracing. Each time when a seed is integrated, a direction is obtained by sampling the local vector distribution.

Therefore any flow line created by each Monte Carlo seed is different from others. These lines can form a bundle, which not only shows a general trend of the particle trace, but also provides an estimate of the regions that the particle has a probability to reach.

The baseline method of Monte Carlo tracing is to integrate the seeds by sampling a vector from the unconditional distribution $g(\vec{v})$ as in Equation 1. Our method samples the vector from the conditional GMM as in Equation 8. We call this *Conditional Monte Carlo* method (CMC for short). Because the conditional GMM used by our method will recompute the weights for the components according to the integration direction in the previous step, a component can have an increased probability to be picked if it fits its corresponding transition pattern better. One detail to mention is that our sampling on the conditional GMM starts from the second step of the tracing. At the first step, since there is no previous direction available, we will just use the unconditional GMM to sample a direction.

4.2 Single Flow Line Tracing

Besides using a bundle of Monte Carlo traces to explore the uncertain vector field, tracing a single streamline or pathline of higher accuracy is also important, which is more suitable when using multiple particle traces together to show flow structures. In a deterministic vector field, the integration direction of the particle is mostly a determinate vector. In a distribution-based flow field, to achieve an integration direction \vec{v}_k from a vector distribution, a good choice is to use the mean vector, because it minimizes the mean squared deviation of samples.

The baseline method uses the mean vector from the unconditional GMM distribution in Equation 1. Our method uses the mean vector of the conditional GMM in Equation 8. We call this *Conditional Mean Vector* method (CMV for short). Similar with the CMC method, the conditional GMM also starts to take effect from the second step. The essential difference between our CMV method and the baseline method is that, it not only exploits the local distribution, but also benefits from the last direction by utilizing the vector transition patterns. At different locations of the data domain, this pattern can be prominent, or relatively vague. When the pattern is not strong enough, the benefit of involving the conditional GMM may not be as clear, so in this situation, we can choose not to use the conditional GMM, but simply use the mean of the original vector GMM to integrate the particle. We choose a threshold for the winding angle thr_θ . Only when the mean of the winding angle distribution has an absolute value larger than thr_θ , we will apply the conditional GMM. In this way the vector transition pattern we adopt always has a strong tendency.

5 RESULT

In this section, we demonstrate two case studies of applying the Conditional Monte Carlo (CMC) method to explore distribution-based vector fields. Then we apply the Conditional Mean Vector (CMV) method, and show the experiment of estimating its error.

5.1 Case Study on Ocean Dataset

The first dataset we apply our method on is the Ocean dataset, which is a simulation dataset that computes the oceanic currents in the Gulf of Mexico and the Caribbean Sea. It has a resolution

of 574×289 with 37 time frames, and is developed by the Climate, Ocean and Sea Ice Modeling program at LANL and Science UV-CDAT team from LLNL.

We first closely look into a small region around a vortex in Figure 2(a). We divide the original dataset using 10×10 blocks, and fit a GMM with two Gaussian components for each block. In the figure, we marked out a block by a black square for detailed illustration. The two Gaussian components of the vector distribution for this block are denoted by two dashed fan-shapes in Figure 2(b), with the red solid arrow representing the mean (normalized for easier illustration). The two components revealed two major vector trends in this block, one going up-left, the other going up-right. The Gaussian components of the winding angle distribution, corresponding to each vector Gaussian component, are shown by the curves on the right side of Figure 2(b). From a selected location, we traced 100 Monte Carlo seeds using the baseline method as the magenta lines in Figure 2(c), and using our CMC method as the magenta lines in Figure 2(d). In Figure 2(c), many traces turn up-left after entering this block, as driven by the Gaussian component with the up-left trend. Eventually, the streamline bundle diverges into two clusters. We select a typical direction by which the tracers entered the block, as shown by the solid green arrows in Figure 2(b). The winding angles formed by the green arrow, indicating the previous integration direction, and the red arrow, indicating the mean direction of each Gaussian component, are marked as θ_1 and θ_2 in the figure. Two green vertical lines are used to mark their locations in the distribution chart. We can see that the θ_2 will have a much higher probability density value, which suggests that the vector transition pattern in the bottom Gaussian component fits the current state much better than the top one. Therefore our method in Figure 2(d) will give bonus weight to this Gaussian component, which causes more traces to follow the up-right trend. The green line in Figure 2(c) and (d) is the streamline traced in the original data using traditional methods as Figure 2(a). We can see that our method can generate Monte Carlo flow lines that better follow the original trend. Note that our method does not intend to fully remove less preferred branches of a bundle. It may still keep a small number of flow lines that travel along these branches, as a natural outcome of the probability model.

Figure 3 gives two more examples of pathline tracing in the reduced Ocean dataset. We also apply the GMM with two Gaussian components, and use a block size 8×8 . In the same way, the magenta lines in the figure are pathlines traced by Monte Carlo seeds, and the green lines are the ones traced in the original dataset. One example in Figure 3(a) shows that the bundle created by the baseline Monte Carlo method encounters several diverging points, as the black squares marked in the figure. These diverging behaviors frequently happen when the underlying flow structures are complex. The bottom of Figure 3(a) shows that using our method, most pathlines turn to the same branch with the original line at the diverging points, and the whole bundle follows the original trend better.

The difference between our method and the baseline method is more obvious when the vector Gaussian components have more disparity, but we also find that the difference is less obvious when the original vector structure is smooth and less complex. Figure 3(b) shows an example where the seeds travel at a relatively stable

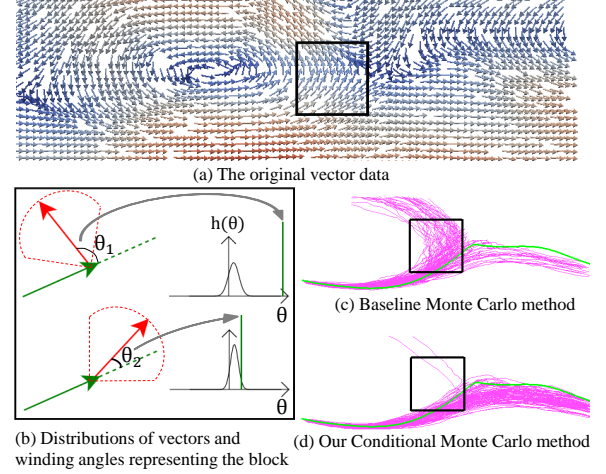


Figure 2: An example of using CMC for streamline tracing. (b) gives the vector and winding angle distribution of the block marked by black squares in other subfigures. In (c)(d), magenta lines: traces by Monte Carlo seeds in the reduced field; green lines: traced in the original data.

region. In this case, both bundles do not have obvious diverging and do not differ obviously.

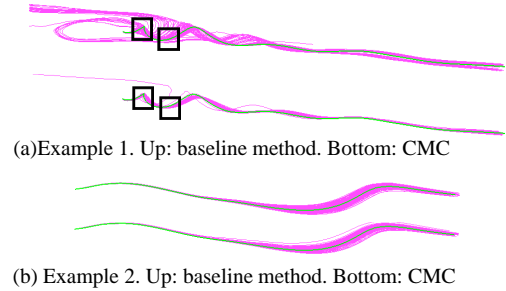


Figure 3: Two examples of pathlines using the Ocean dataset. Traces are colored in the same way with Figure 2.

5.2 Case Study on Eddy Dataset

The second dataset we processed is the Eddy dataset. It is a dataset of an eddy resolving simulation of the oceans in the whole world. Its resolution is 3600×2400 , for a total number of 36 time frames. We divide the original dataset using 16×16 blocks, and fit a GMM with three Gaussian components for each block. We also used 100 Monte Carlo seeds to generate the trace bundle.

Figure 4 gives an example of streamline tracing from a location at the north Pacific Ocean. We can see that the complex ocean structure makes the baseline method fail to capture the original streamline, and all Monte Carlo seeds eventually deviate far away. We detect a block as marked by the black square in the figure. As in Figure 4(c), the red solid arrows represent the normalized mean of three Gaussian components of the vector GMM. While the actual streamline is supposed to go downwards at the block, most

Monte Carlo traces are driven leftwards by the first and the second Gaussian components. The green solid arrow in Figure 4(c) is a typical direction by which the tracers entered the block. It forms a winding angle with the mean of each Gaussian component as θ_1 , θ_2 , and θ_3 in the figure. When taking this previous direction into account, we can see that θ_3 has a much higher probability density in the corresponding angle distribution. Therefore the conditional GMM will reward the third Gaussian component by an increased weight. As a result, when using our method, many streamlines will turn downwards, which correctly follows the original trend.

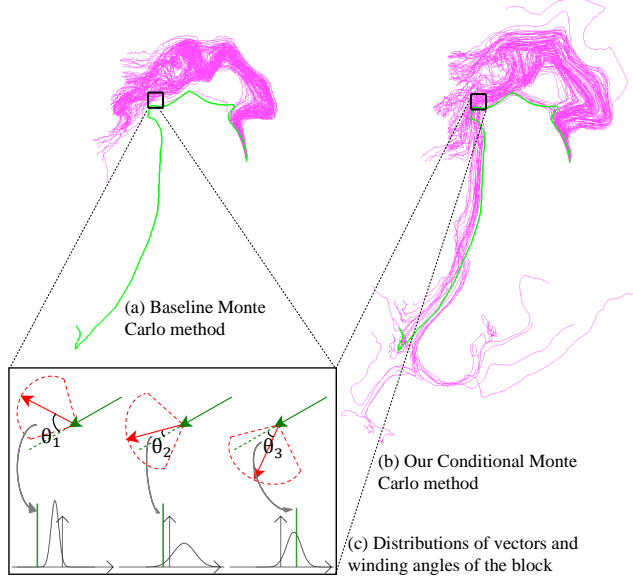


Figure 4: A streamline tracing example in the Eddy dataset. Traces are colored in the same way with Figure 3. (c) shows the vector and winding angle distributions of the block marked by the black square in (a) and (b).

The above example also exhibits the limitation of the idea used in previous research [Friman et al. 2006; Malcolm et al. 2010; Zhang et al. 2009], which does not compute the quantified vector transition, but always rewards the direction that is closer to the previous direction. In this example, if we adopt the same policy, the second Gaussian component will become the most rewarded one. This will still drive a lot of particles leftwards, and cannot follow the original trend as close as our method.

5.3 Single Flow Line Error Estimation

In this section, we show results of applying the Conditional Mean Vector method to trace a single streamline or pathline in the reduced Ocean and Eddy datasets, and make comparisons with the baseline method. The number of the Gaussian components for each block of the Ocean dataset was two, and was three for the Eddy dataset, with various block sizes tested. We placed $24 \times 12 = 288$ particles in each reduced Ocean dataset, and $45 \times 30 = 1350$ particles in each reduced Eddy dataset, uniformly distributed over the data domain. From each position we traced a streamline on the first time frame, and a pathline using all time frames, using both our CMV method and the

Table 1: The average error of our CMV methods E_{CMV} , comparing with the error of the baseline method E_{MV} .

Dataset	Block Size	Streamline		Pathline	
		E_{CMV}	E_{MV}	E_{CMV}	E_{MV}
Ocean	8×8	1.75371	1.76867	11.989	13.9853
	16×16	4.74736	5.01662	31.9457	32.8427
Eddy	16×16	17.8619	18.1388	58.351	58.5562
	32×32	22.169	22.2099	63.6412	63.9901

baseline mean vector (MV) method. The purpose of this task is to recover the flow lines that would be traced in the original datasets. Therefore, we also traced these particles in the original datasets from the same location, and those traces will serve as the tracing targets. For each flow line traced in a reduced dataset from a particle p_i by method M , its error $e_M(p_i)$ is measured by its distance with the tracing target achieved in the original dataset. For a streamline, the distance is measured by the widely applied Mean-of-Closest distance metric [Corouge et al. 2004; Mobergs et al. 2005; Rössl and Theisel 2012; Tao et al. 2013; Zhang et al. 2008]. For a pathline, because the nodes along the trace are directly parameterized by time, we measure the error by the deviation of the particle location at the last time step. The average error E_M for method M is then defined as the average of the errors of all particles.

The error of our CMV method E_{CMV} and the baseline method E_{MV} are shown in Table 1. The unit of the average error is the grid cell of the raw data. We can see that our method has lower average error than the baseline method for all tested cases, which shows that flow lines generated using our method are averagely closer to the corresponding flow lines in the original datasets. We want to mention that the target flow lines themselves are also mathematical approximations. We do not claim that our method is more accurate compared to any ground truth, but demonstrate its effectiveness in preserving and recovering the lost information. Also the advantage is in an average manner, instead of for every single particle.

Figure 5 gives an example of streamline tracing from a particle in the Ocean dataset reduced by 8×8 . The green line is the tracing target. The blue line is the trace generated using our method, while the red line is the trace generated using the baseline method. The traces of our method and the baseline method are overlapped at early stages, because the particle is at stable regions where the average winding angle is less than the selected threshold. Taking a block marked by the black square as example, the solid green arrow indicates the direction used in the last step by our method, which forms winding angles θ_1 and θ_2 with the mean of each Gaussian component. Since for the first Gaussian component, the winding angle has a much higher probability density in the angle distribution, so when computing the conditional mean vector at the current step, the weight of this component will be increased. This leads to our trace going more leftwards, and staying closer to the target.

6 DISCUSSION

6.1 Performance

Compared to baseline methods that only use unconditional GMM distributions, our method does bear some overhead regarding to computational time and storage. In this section, we show that the overhead is relatively small.

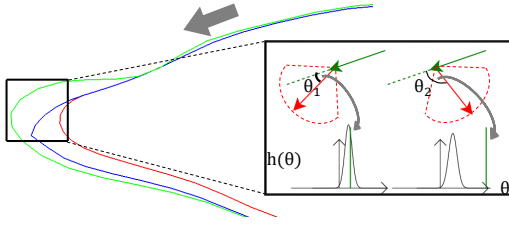


Figure 5: A streamline tracing example. Green line: the tracing target. Blue line: CMV. Red line: baseline method.

Table 2: Comparisons of computation time (in seconds) for our methods and the baseline methods.

	Data Reduction		Single Pathline Tracing		Monte Carlo Tracing	
	Baseline	Our Method	Baseline	Our Method	Baseline	Our Method
Time	73.35	76.53	0.1003	0.1080	3.307	5.480

Table 2 shows several collected key time costs on a regular desktop computer, using the Ocean dataset, with a 8×8 block size and two Gaussian components for each block. The time cost to perform data reduction, i.e., extracting distributions from the raw data, does not increase substantially. The most costly part of the reduction is the large number of iterations used by the EM algorithm. When our method performs the extra computation for the distribution of winding angles, as shown in Section 3.2, we reuse the probability $P_i(j)$ of the vector GMM, and there is no necessity to redo the loop of the EM algorithm again. The extra time needed for flow line tracing is for estimating the conditional vector distribution, and it must be performed on the fly because the direction of the last step cannot be known in advance. Compared to the baseline algorithm, the computation time of our algorithm for tracing a single flow line (the CMV method) does not increase much, because we apply the angle threshold that exempts us from performing the conditional distribution estimation when the vector transition does not have a strong pattern. Many real scientific simulations contain large area of smooth regions where the threshold is not reached. The time cost for the CMC method does increase obviously, but we believe the better results that we can achieve are still worth the cost.

For each multivariate Gaussian component of vectors, the basic GMM model needs to store one number for the weight, two numbers for the mean, and three numbers for the symmetric 2×2 covariance matrix. Our model adds two numbers for the winding angle mean and variance. So the space cost increases by 33.3% compared to basic GMM. Even though, it still saves a lot of space than the original dataset when it is necessary to perform data reduction.

6.2 Parameter Analysis

One parameter involved in our method is the step size L used to compute the winding angle during the reduction phase. Selecting a proper step size itself is an important issue for particle tracing, but it is not the focus of our study. All the demonstrated results in Section 5 were generated using $L = 0.25$ and $l = 0.25$, in the unit of the interval between time frames. We tested various combinations of L and l from the set $\{0.125, 0.25, 0.5\}$, and were able to reproduce similar visual improvement and error decrease. However,

in extreme cases such as when L and l differ too much, the linear compensation conducted in Section 4 may fail. A more through study is expected as a future work.

Another parameter used for the CMV method is the threshold for the winding angle thr_θ . We repeated the pathline tracing experiment in the 8×8 reduced Ocean dataset in Section 5.3 with different choices of thr_θ . As in Figure 6, performing the CMV method without a threshold may not always achieve the lowest error. This is because at regions with low winding angles, mostly the baseline method already performs very good; on the other hand introducing additional estimation can bring extra error. Generally as thr_θ increases, the error will go down first until reaching an optimal threshold. Then it will go up and converge, since with increased thresholds, less and less integration steps will adopt the conditional distribution estimation. Finding the optimal value is data dependent and is not easy. However, users can always select a relatively large threshold, since as we have demonstrated that our method is most useful on unstable regions with larger winding angles.

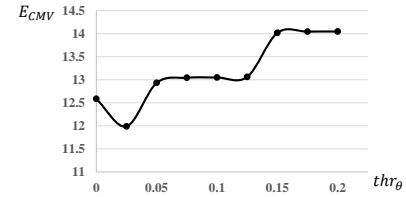


Figure 6: For the CMV method, the relation of the winding angle threshold thr_θ and the average tracing error.

7 CONCLUSION

In this paper, we proposed flow line tracing methods in distribution-based vector fields, by utilizing distributions of winding angles. When performing flow line tracing, the winding angle distribution helped derive a conditional distribution of local vectors, affected by the integration direction used in the last step. We used case studies and experiments to demonstrate that the winding angle distribution is a valuable tool for distribution-based vector field visualization.

One limitation of our method is as shown in Figure 3(b), our results do not have significant differences with the baseline methods where the vector transition pattern is not evident. Nevertheless, those regions are seldom a major interest of scientists, because they contain very few complex flow structures or variable trends. So our method can carry a value in more interesting regions.

Our future work is to extend the current method to 3D flow datasets. Also besides the current application on data reduction, we will exploit the benefits from the winding angle in other uncertain visualization applications, such as for ensemble field analysis.

ACKNOWLEDGMENTS

This work was supported in part by NSF grants IIS-1250752, IIS-1065025, and US Department of Energy grants DE-SC0007444, DE-DC0012495, program manager Lucy Nowell.

REFERENCES

- Tushar Athawale and Alireza Entezari. 2013. Uncertainty Quantification in Linear Interpolation for Isosurface Extraction. *Visualization and Computer Graphics, IEEE Transactions on* 19, 12 (2013), 2723–2732.
- Tushar Athawale, Elham Sakhaee, and Alireza Entezari. 2016. Isosurface Visualization of Data with Nonparametric Models for Uncertainty. *Visualization and Computer Graphics, IEEE Transactions on* 22, 1 (2016), 777–786.
- Jeff A Bilmes et al. 1998. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. *International Computer Science Institute* 4, 510 (1998), 126.
- Ralf P Botchen, Daniel Weiskopf, and Thomas Ertl. 2005. Texture-based visualization of uncertainty in flow fields. In *Visualization, 2005. VIS 05. IEEE*. IEEE, 647–654.
- Kai Bürger, Roland Fraedrich, Dorit Merhof, and Rüdiger Westermann. 2012. Instant visitation maps for interactive visualization of uncertain particle trajectories. In *IS&T/SPIE Electronic Imaging*. 82940P–82940P.
- National Hurricane Center. 2016. Definition of the NHC Track Forecast Cone. (2016). <http://www.nhc.noaa.gov/aboutcone.shtml>.
- Abon Chaudhuri, Teng-Yok Lee, Bo Zhou, Cong Wang, Tiantian Xu, Han-Wei Shen, Tom Peterka, and Yi-Jen Chiang. 2012. Scalable computation of distributions from large scale data sets. In *Large Data Analysis and Visualization (LDAV), IEEE Symposium on*. 113–120.
- Isabelle Corouge, Sylvain Gouttard, and Guido Gerig. 2004. Towards a shape model of white matter fiber bundles using diffusion tensor MRI. In *Biomedical Imaging: Nano to Macro, IEEE International Symposium on*. 344–347.
- Jonathan Cox, Donald House, and Michael Lindell. 2013. Visualizing uncertainty in predicted hurricane tracks. *International Journal for Uncertainty Quantification* 3, 2 (2013).
- Soumya Dutta, Chun-Ming Chen, Gregory Heinlein, Han-Wei Shen, and Jen-Ping Chen. 2017. In Situ Distribution Guided Analysis and Visualization of Transonic Jet Engine Simulations. *IEEE transactions on visualization and computer graphics* 23, 1 (2017), 811–820.
- Soumya Dutta and Han-Wei Shen. 2016. Distribution driven extraction and tracking of features for time-varying data analysis. *IEEE transactions on visualization and computer graphics* 22, 1 (2016), 837–846.
- Florian Ferstl, Kai Bürger, and Rüdiger Westermann. 2016. Streamline Variability Plots for Characterizing the Uncertainty in Vector Field Ensembles. *Visualization and Computer Graphics, IEEE Transactions on* 22, 1 (2016), 767–776.
- Ola Friman, Gunnar Farneback, and Carl-Fredrik Westin. 2006. A Bayesian approach for stochastic white matter tractography. *Medical Imaging, IEEE Transactions on* 25, 8 (2006), 965–978.
- Hanqi Guo, Xiaoru Yuan, Jian Huang, and Xiaomin Zhu. 2013. Coupled ensemble flow line advection and analysis. *Visualization and Computer Graphics, IEEE Transactions on* 19, 12 (2013), 2733–2742.
- Mihaela Jarema, Ismail Demir, Johannes Kehrner, and Rüdiger Westermann. 2015. Comparative visual analysis of vector field ensembles. In *Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on*. IEEE, 81–88.
- Michaela Jarema, Johannes Kehrner, and Rüdiger Westermann. 2016. Comparative visual analysis of transport variability in flow ensembles. (2016).
- Le Liu, Mahsa Mirzargar, Robert M Kirby, Ross Whitaker, and Donald H House. 2015. Visualizing Time-Specific Hurricane Predictions, with Uncertainty, from Storm Path Ensembles. In *Computer Graphics Forum*, Vol. 34. 371–380.
- Shusen Liu, Joshua A Levine, P Bremer, and Valerio Pascucci. 2012. Gaussian mixture model based volume visualization. In *Large Data Analysis and Visualization (LDAV), IEEE Symposium on*. 73–77.
- Alison L Love, Alex Pang, and David L Kao. 2005. Visualizing spatial multivalued data. *Computer Graphics and Applications, IEEE* 25, 3 (2005), 69–79.
- James G Malcolm, Oleg Michailovich, Sylvain Bouix, Carl-Fredrik Westin, Martha E Shenton, and Yogesh Rathi. 2010. A filtered approach to neural tractography using the Watson directional function. *Medical Image Analysis* 14, 1 (2010), 58–69.
- Mahsa Mirzargar, Ross T Whitaker, and Robert M Kirby. 2014. Curve boxplot: Generalization of boxplot for ensembles of curves. *Visualization and Computer Graphics, IEEE Transactions on* 20, 12 (2014), 2654–2663.
- Bart Moberts, Anna Vilanova, and Jarke J Van Wijk. 2005. Evaluation of fiber clustering methods for diffusion tensor imaging. In *Visualization, IEEE*. 65–72.
- Mathias Otto, Tobias Germer, Hans-Christian Hege, and Holger Theisel. 2010. Uncertain 2D vector field topology. In *Computer Graphics Forum*, Vol. 29. 347–356.
- Mathias Otto, Tobias Germer, and Holger Theisel. 2011. Uncertain topology of 3D vector fields. In *Pacific Visualization Symposium (PacificVis), IEEE*. 67–74.
- Christoph Petz, Kai Pöthkow, and Hans-Christian Hege. 2012. Probabilistic local features in uncertain vector fields with spatial correlation. In *Computer Graphics Forum*, Vol. 31. 1045–1054.
- Tobias Pfaffelmoser, Matthias Reitering, and Rüdiger Westermann. 2011. Visualizing the positional and geometrical variability of isosurfaces in uncertain scalar fields. In *Computer Graphics Forum*, Vol. 30. 951–960.
- Kai Pöthkow and Hans-Christian Hege. 2011. Positional uncertainty of isocontours: Condition analysis and probabilistic measures. *Visualization and Computer Graphics, IEEE Transactions on* 17, 10 (2011), 1393–1406.
- Kai Pöthkow, Britta Weber, and Hans-Christian Hege. 2011. Probabilistic marching cubes. In *Computer Graphics Forum*, Vol. 30. 931–940.
- Paula Ceccon Ribeiro, Haroldo Fraga de Campos Velho, and Hélio Lopes. 2016. Helmholtz–Hodge decomposition and the analysis of 2D vector field ensembles. *Computers & Graphics* 55 (2016), 80–96.
- Christian Rössl and Holger Theisel. 2012. Streamline embedding for 3D vector field exploration. *Visualization and Computer Graphics, IEEE Transactions on* 18, 3 (2012), 407–420.
- Kuangyu Shi, Holger Theisel, Helwig Hauser, Tino Weinkauff, Kresimir Matkovic, Hans-Christian Hege, and Hans-Peter Seidel. 2009. 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*. Springer, 75–88.
- Jun Tao, Jun Ma, Chaoli Wang, and Ching-Kuang Shene. 2013. A unified approach to streamline selection and viewpoint selection for 3D flow visualization. *Visualization and Computer Graphics, IEEE Transactions on* 19, 3 (2013), 393–406.
- David Thompson, Joshua A Levine, Janine C Bennett, P-T Bremer, Attila Gyulassy, Valerio Pascucci, and Philippe P Pébay. 2011. Analysis of large-scale scalar data using hixels. In *Large Data Analysis and Visualization (LDAV), IEEE Symposium on*. 23–30.
- Ko-Chih Wang, Kewei Lu, Tzu-Hsuan Wei, Naeem Shareef, and Han-Wei Shen. 2017. Statistical Visualization and Analysis of Large Data Using a Value-based Spatial Distribution. In *Pacific Visualization Symposium (PacificVis), IEEE (in press)*.
- Fan Zhang, Edwin R Hancock, Casey Goodlett, and Guido Gerig. 2009. Probabilistic white matter fiber tracking using particle filtering and von Mises–Fisher sampling. *Medical Image Analysis* 13, 1 (2009), 5–18.
- Song Zhang, Stephen Correia, and David H Laidlaw. 2008. Identifying white-matter fiber bundles in DTI data using an automated proximity-based fiber-clustering method. *Visualization and Computer Graphics, IEEE Transactions on* 14, 5 (2008), 1044–1053.