

eFESTA: Ensemble Feature Exploration with Surface Density Estimates

Wenbin He, Hanqi Guo, *Member, IEEE*, Han-Wei Shen, *Member, IEEE*, and Tom Peterka *Member, IEEE*,

Abstract—We propose *surface density estimate* (SDE) to model the spatial distribution of surface features—isosurfaces, ridge surfaces, and streamsurfaces—in 3D ensemble simulation data. The inputs of SDE computation are surface features represented as polygon meshes, and no field datasets are required (e.g., scalar fields or vector fields). The SDE is defined as the kernel density estimate of the infinite set of points on the input surfaces and is approximated by accumulating the surface densities of triangular patches. We also propose an algorithm to guide the selection of a proper kernel bandwidth for SDE computation. An *ensemble Feature Exploration* method based on *Surface densiTy EstimAtes* (eFESTA) is then proposed to extract and visualize the major trends of ensemble surface features. For an ensemble of surface features, each surface is first transformed into a density field based on its contribution to the SDE, and the resulting density fields are organized into a hierarchical representation based on the pairwise distances between them. The hierarchical representation is then used to guide visual exploration of the density fields as well as the underlying surface features. We demonstrate the application of our method using isosurface in ensemble scalar fields, Lagrangian coherent structures in uncertain unsteady flows, and streamsurfaces in ensemble fluid flows.

Index Terms—Density estimation, ensemble data visualization, uncertainty visualization, feature exploration.

1 INTRODUCTION

ENSEMBLE simulations are becoming prevalent in various scientific and engineering domains, such as climate, weather, aerodynamics, and computational fluid dynamics. An ensemble is a collection of data produced by simulations for the same physical phenomenon conducted with different initial conditions, parameterizations, or phenomenological models. Ensemble simulations are used to simulate complex systems, study sensitivities to initial conditions and parameters, and mitigate uncertainty. For example, in numerical weather prediction, ensemble forecasts with different forecast models and initial conditions are widely used to indicate the range of possible future states of the atmosphere.

The focus of this work is modeling and visualizing the positional uncertainty of surface features extracted from 3D ensemble simulation data. Surface features such as isosurfaces, ridge surfaces, and streamsurfaces play an important role in many scientific applications. Uncertainty quantification of surface features based on polygon meshes instead of field datasets (e.g., scalar fields or vector fields) is important because field datasets that can be used to quantify the positional uncertainty of surface features are often not available. For example, for specific applications such as ridge surfaces and streamsurfaces, the positional uncertainty of the surfaces can not be quantified based

on the vector fields without known the geometries of the surfaces. Only using the field data, the application is limited to modeling the positional uncertainty of isosurfaces, for which various techniques [1], [2], [3], [4], [5], [6], [7], [8], [9] have been proposed. Moreover, the majority of the previous techniques [1], [2], [6], [7], [8], [9] model the uncertainty of the scalar values at each grid point as an independent random variable without considering correlations between the random variables at different spatial locations. Only a few techniques [3], [4], [5] consider the correlations of the random variables to generate more reliable results. In this work, isosurfaces are extracted from each ensemble member independently, such that spatial correlations of scalar values within each ensemble member are preserved.

Given an ensemble of surfaces, a straightforward density estimation approach is to define a regular grid over the surfaces, and then count the number of surfaces intersecting each grid cell. However, after discretizing the surfaces with respect to a given grid, the information of the surface patches (e.g., location, orientation, and shape) within each grid cell is lost, which introduces discretization error into the density estimation results. Although increasing the grid resolution can reduce the discretization error, the computation cost increases. In this work, we propose *surface density estimate* (SDE), which generalizes the kernel density estimate (KDE) from discrete sample points to the infinite set of points on input surfaces. We approximate SDE of the input surfaces by accumulating the surface densities of triangular patches, which can be calculated based on bivariate normal integrals with efficient GPU computation.

We also propose an algorithm to guide the selection of a proper kernel bandwidth for SDE computation. Like KDE, the most important parameter for computing a SDE is the bandwidth, which determines the degree of smoothing induced. An improper bandwidth may cause under-

- Wenbin He is with the Department of Computer Science and Engineering, the Ohio State University, Columbus, OH 43210, USA.
E-mail: he.495@osu.edu
- Hanqi Guo is with the Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL 60439, USA.
E-mail: hguo@anl.gov
- Han-Wei Shen is with the Department of Computer Science and Engineering, the Ohio State University, Columbus, OH 43210, USA.
E-mail: shen.94@osu.edu
- Tom Peterka is with the Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL 60439, USA.
E-mail: tpeterka@mcs.anl.gov

oversmoothing problems. Our bandwidth selection method is based on the variability of input surfaces. We extend the least squares cross validation (LSCV) method to approximate the mean integrated squared error (MISE) between the SDE and the target density field. The bandwidth is then selected by minimizing the approximated MISE.

To further extract and visualize the major trends of ensemble surface features, we develop an efficient interactive visualization framework called *eFESTA* (*e*nsemble *F*eature *E*xploration based on *S*urface *densi**T*y *E*stim*A*tes). In this framework, surface of each ensemble member is first transformed into a density field based on its contribution to the SDE. The density fields are then organized into a hierarchical representation based on their pairwise distances. The hierarchical representation is then used to guide the visual exploration of the major trends of the underlying surfaces.

We compare the accuracy and performance of our density estimation method with alternative approaches using synthetic uncertain surfaces that have known spatial distributions. Our method is able to generate more accurate density estimation results when the computation cost is similar to the alternative approaches. We also demonstrate the effectiveness and usefulness of the proposed method for different applications including visualizing and analyzing variabilities of isosurfaces in ensemble simulations, Lagrangian coherent structures (LCSs) [10] in uncertain unsteady flows, and streamsurfaces in ensemble fluid flows. For the application of isosurfaces in ensemble simulations, the proposed method generates comparable results to other state-of-the-art approaches that rely on scalar fields. Our method can also handle applications that are depend on extracted surface features such as LCSs and streamsurfaces. Moreover, by applying similarity guided visual exploration, the major trends of ensemble surfaces can be extracted and visualized effectively through user interaction. In summary, the contributions of this study are threefold:

- A density estimation method to visualize and analyze ensemble surfaces
- An algorithm to guide the selection of kernel bandwidth for SDE computation
- A visual exploration framework for the derived density fields

2 RELATED WORK

We review related works in ensemble visualization, uncertainty visualization, and density estimation.

2.1 Ensemble Visualization

Ensemble visualization techniques can be categorized into location- and feature-based approaches [11], [12]. Location-based approaches compare ensemble properties at fixed locations in the domain. Feature-based approaches extract features from individual ensemble members and then compare them across the ensemble.

We regard our study as a feature-based ensemble visualization approach, and we review closely related studies on feature-based ensemble visualization. Spaghetti plots are a well-known visual analysis technique that overlays an ensemble of features such as isocontours to compare

among ensemble members [13]. Phadke et al. [14] proposed two ensemble visualization methods, pairwise sequential animation and screen door tinting, for visualizing ensembles that contain numerous members. Sanyal et al. [15] proposed a tool named Noodles that uses glyphs and confidence ribbons to visualize uncertainties for operational meteorologists. Whitaker et al. [16] and Mirzargar et al. [17] proposed contour and curve box-plots, respectively, to visualize ensembles of curves based on the concept of statistical band depth. Raj et al. [18] extended contour box-plots to 3D for brain atlas construction and analysis. Zehner et al. [19] proposed a method to render multiple isosurfaces with a specific colour scheme. Ferstl et al. [20] presented a technique to explore the major trends in an ensemble of streamlines based on principal component analysis and hierarchical clustering. More recently, Ferstl et al. [21] presented an approach that uses signed distance functions to generate contour variability plots for the visual analysis of an ensemble of isocontours. They further extended the approach for visualizing the spatial and temporal evolution of isocontours in ensembles of 2D time-varying scalar fields [22]. Demir et al. [23] proposed a technique to visualize ensembles of isosurfaces based on screen-space silhouettes.

However, the previous works mainly focus on line features (e.g. contour lines); and for surface features the previous works can not show point-wise quantitative information. Unlike previous works, we analyze the variability of ensemble surfaces using density estimation, thus the probability density of surfaces passing through any given location can be quantified.

2.2 Uncertainty Visualization

We summarize the closely related studies on uncertainty quantification and visualization of features extracted from uncertain scalar and vector datasets, such as isocontours and flow lines. Comprehensive reviews of uncertainty visualization can be found in [24], [25], [26].

Uncertain scalar field visualization For visualizing and analyzing features extracted from uncertain scalar datasets, such as isocontours, a number of visualization techniques have been presented. A class of uncertainty visualization techniques conveys uncertainties by visualizing potential errors or variability of point positions with colors [27], [28], [29] or point-based primitives [30]. Recent studies focus more on uncertainty quantification and visualization using statistical analysis techniques. Thompson et al. [1] introduced a new data representation for uncertain scalar data, called hixels, which enables fuzzy isosurface visualization to indicate the possible isosurface locations in the data. Pöthkow and Hege [2] proposed a technique that approximates the level crossing probabilities (LCPs) [5] with parametric models for uncertainty analysis of isocontours. Based on this method, Pöthkow et al. proposed the probabilistic marching cubes algorithm [4]. Recently, Pöthkow and Hege proposed a nonparametric statistical analysis framework for uncertain isosurfaces visualization [31]. Athawale and Entezari [6] presented a closed-form computation of the level crossing probabilities for studying the interaction between linear interpolation and data uncertainty quantified by the uniform distribution. More recently, Athawale et al. presented an isosurface

extraction algorithm [7] for uncertain scalar fields where the data uncertainty is modeled by nonparametric statistics. Hazarika et al. [32] proposed a coupla-based technique for LCP approximation. Pfaffelmoser et al. proposed [3] a method to determine more reliable LCPs by considering correlations in the data. Kumpf et al. proposed contour probability plots [9] to depict lobes, which indicate the probability that a contour line is locally contained in the lobe. Sakhaee and Entezari [8] presented a spline-based framework to quantify and propagate the uncertainty through the volume rendering pipeline.

In general, the previous works mainly focused on modeling the positional uncertainty of isosurfaces using uncertain scalar fields. Unlike the previous works, we take a geometry based approach that analyzes and visualizes variability of surfaces represented as geometries (e.g., triangular meshes), such that our method can be applied on applications that scalar fields are unavailable (e.g., LCSs and streamsurfaces).

Uncertain vector field visualization Various techniques have been proposed for visualizing uncertain vector datasets. A class of techniques focus on encoding uncertainties in vector datasets as additional visual channels, such as glyphs [33], [34], [35] and textures [36]. Another class of techniques analyze and visualize features extracted from uncertain vector datasets by extending methods used in deterministic flow fields such as streamlines and stream ribbons [24]. Petz et al. [37] presented a statistical analysis framework to extract probabilistic local features from uncertain vector fields considering their spatial correlation structure. Bhatia et al. [38] presented streamwaves to visualize spatial errors of edge maps used in particle tracing. The concept of LCS has also been extended to visualize and analyze uncertain time-varying flow fields by Haller [39] and Guo et al. [40]. Unlike previous methods that mainly focus on studying uncertain local features or flow lines, the proposed method in this paper can be applied to uncertain surface features such as LCSs and streamsurfaces extracted from uncertain flow fields.

2.3 Density Estimation

We review studies on density estimation for various applications in the context of data analysis and visualization, including density estimation of particles, graphs, parallel coordinate plots (PCPs), trajectories, and surfaces.

Density estimation of particles has been extensively studied for visualization and analysis. Peterka et al. [41] reviewed and evaluated density estimation methods that transform particles to a probability density field, including cloud in cell (CIC) [42], smoothed particle hydrodynamics (SPH) [43], tessellation (TESS) [44], [45], and adaptive cloud in cell (ACIC) [46]. CIC [42] is a first-order method that linearly interpolates the particle's mass to points of estimation within a fixed-size hypercube. The size and shape of the region over which the particle's mass is distributed were further adjusted for each particle in the SPH [43] and TESS [44], [45]. Concepts of CIC and SPH were then combined in ACIC [46], in which the size of the hypercube adapts to cover a given number of particles as in SPH.

Density estimation has also been extended to analyze and visualize more complex data types, including

graphs, PCPs, trajectories, and surfaces. Zinsmaier et al. [47] proposed a technique that aggregates node density with KDE [48] and generates meta edges for visualizing large graphs interactively with adjustable levels of detail. Hurter et al. [49] proposed a method to compute bundled layouts of general graphs based on density estimates of graph edges. Heinrich and Weiskopf [50] proposed continuous parallel coordinates, which visualizes PCPs using density estimation. Zhou et al. [51] proposed a method to visualize PCPs as density fields using splatting. Muigg [52] employed anisotropic diffusion of noise textures to visualize line orientations for PCPs. Scheepens et al. presented a method to explore density maps of trajectories with multiple attributes [53] and extended it by compositing multiple density fields for more comprehensive analysis [54]. Lampe et al. presented KDE-based visual analysis methods for scatterplots of dynamic data [55] and spatiotemporal trajectories [56]. They further presented a technique for rendering smooth curves independent of frequencies, zoom level, and models based on KDE [57]. Guo et al. [40] proposed a method to transform uncertain LCSs into a density field based on SPH and a zero-order kernel function that is uniformly distributed in the volume of a sphere. Compared with U-LCS, our work generalizes kernel density estimation with higher-order kernel functions (e.g., the Gaussian kernel) for transforming ensemble surfaces to density fields.

3 OVERVIEW

In this work, we start with an ensemble of surface features represented as polygon meshes. Instead of visualizing the surfaces directly, such as with overlaid rendering, we visualize and analyze the surfaces based on their spatial distribution. To model the spatial distribution of the surfaces, we present *surface density estimate* (SDE) (details in Section 4), which we define as the kernel densities of the infinite set of points on the input surfaces. By computing the SDE, surface features are transformed into a density field, which is then visualized using existing volume exploration techniques such as volume rendering. A bandwidth selection method (details in Section 5) is also proposed to guide the selection of a proper kernel bandwidth.

To further extract and visualize the major trends of ensemble surface features based on the SDE, we present an efficient interactive visualization framework called *eFESTA* (details in Section 6). Our system has two major modules: the density estimation module and the exploration module, as shown in Figure 1. Given an ensemble of surfaces S_0, S_1, \dots, S_{N-1} (Figure 1(a)), we first select a bandwidth for SDE computation based on the variability of the surfaces. Then for each surface S_i , a density field D_i (Figure 1(b)) is generated based on its contribution to the SDE, where the density value at each grid point represents the probability density of the point being on the current surface. The resulting density fields from the density estimation module are then used in the exploration module for comparative analysis and interactive visualization of the underlying surfaces. For efficient visual exploration, we organize the density fields into a hierarchical representation (Figure 1(c)) that groups similar density fields into clusters. This representation is then used

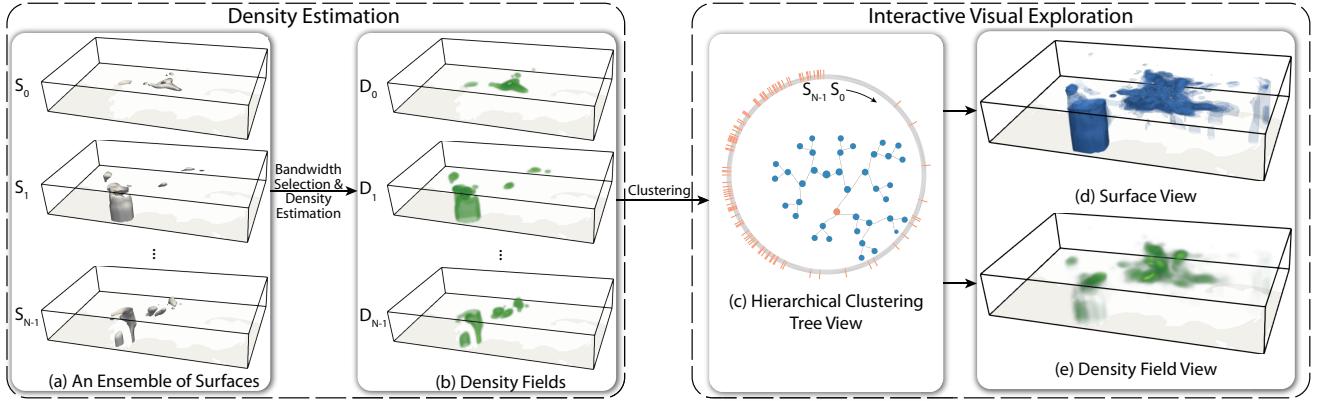


Fig. 1. Overview of the exploration workflow. We begin with an ensemble of surface features (a), whose corresponding density fields (b) are generated based on density estimation. The resulting density fields are organized into a hierarchical representation (c) based on their pairwise distances. The hierarchical representation is then used to guide the visual exploration of the surfaces (d) and the density fields (e).

to guide the visual exploration of the surfaces (Figure 1(d)) and the corresponding density fields (Figure 1(e)).

4 SURFACE DENSITY ESTIMATE

We present SDE, which generalizes KDE from discrete sample points to surfaces. In this section, we first give a formal definition of SDE and then describe how to approximate the SDE from a given surface.

4.1 SDE Definitions

We start with a review of multidimensional KDE for discrete data points and then extend KDE for SDE. Given a set of discrete data points $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{n-1}$ in \mathbb{R}^d , where d is the dimensionality of space, the KDE $\hat{f}_{\mathbf{H}}(\mathbf{x})$ is defined as a convolution of the data points with respect to a smoothing kernel $K_{\mathbf{H}}$ at position \mathbf{x} ,

$$\hat{f}_{\mathbf{H}}(\mathbf{x}) = \frac{1}{n} \sum_{i=0}^{n-1} K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i), \quad (1)$$

where $K_{\mathbf{H}}$ is defined as

$$K_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-\frac{1}{2}} K(\mathbf{H}^{-\frac{1}{2}} \mathbf{x}), \quad (2)$$

\mathbf{H} is a symmetric positive definite bandwidth matrix that controls the degree and orientation of smoothing, and K is the kernel function, which is a symmetric multivariate probability density function. Among various of kernel functions, the multivariate Gaussian kernel,

$$K(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d}} e^{-\frac{1}{2}\mathbf{x}^T \mathbf{x}}, \quad (3)$$

is the most widely used kernel function for KDE, we thus use the multivariate Gaussian kernel to demonstrate the concept of SDE in this study.

Given a surface S , we define the SDE as the kernel densities of the infinite set of points on the surface. In other words, the SDE is the convolution of every point $\mathbf{p} \in S$ with respect to a smoothing kernel $K_{\mathbf{H}}$ at any position \mathbf{x} in \mathbb{R}^3 ,

$$\hat{S}_{\mathbf{H}}(\mathbf{x}) = \frac{1}{Area(S)} \iint_S K_{\mathbf{H}}(\mathbf{x} - \mathbf{p}) dS, \quad (4)$$

where $Area(S)$ is the area of the surface S .

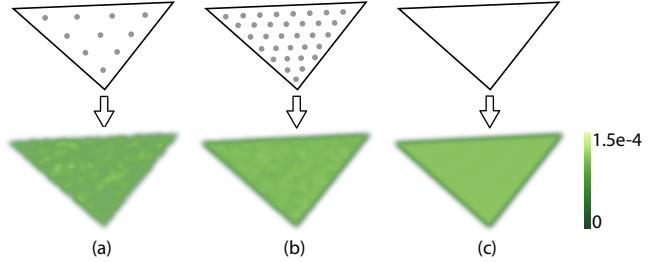


Fig. 2. Density fields generated for a triangular patch: (a) and (b) show KDEs calculated based on 10^4 and 10^5 sample points, respectively; (c) shows SDE calculated analytically based on the triangular patch.

4.2 SDE Calculation

To calculate the SDE for a given surface S , we first discretize the surface as a triangular mesh, which comprises a set of triangular patches T_0, T_1, \dots, T_{n-1} . Then, for each triangular patch T , we compute its surface density $\tau_{\mathbf{H}}(\mathbf{x}, T)$ with respect to the surface S . The $\tau_{\mathbf{H}}(\mathbf{x}, T)$ is defined as the convolution of every point $\mathbf{p} \in T$ with respect to a smoothing kernel $K_{\mathbf{H}}$ at target position \mathbf{x} and normalized by the area of the surface S ,

$$\begin{aligned} \tau_{\mathbf{H}}(\mathbf{x}, T) &= \frac{1}{Area(S)} \iint_T K_{\mathbf{H}}(\mathbf{x} - \mathbf{p}) dT \\ &= \frac{1}{Area(S)} |\mathbf{H}|^{-\frac{1}{2}} \iint_T K(\mathbf{H}^{-\frac{1}{2}} (\mathbf{x} - \mathbf{p})) dT. \end{aligned} \quad (5)$$

We then approximate the SDE of the surface S by accumulating the surface densities contributed by its triangular patches,

$$\hat{S}_{\mathbf{H}}(\mathbf{x}) = \sum_{i=0}^{n-1} \tau_{\mathbf{H}}(\mathbf{x}, T_i), \quad (6)$$

The surface density of a triangular patch can be approximated with the KDE over points sampled from the triangular patch. Figures 2 (a) and (b) show two example density fields constructed for a single triangular patch as KDEs over 10^4 and 10^5 sample points, respectively. More accurate results can be generated by increasing the number of sample points, but the computation cost increases. In this work, we present a method to compute SDE analytically instead. The density field constructed for the same triangular patch using our method is shown in Figure 2 (c). We can see that our

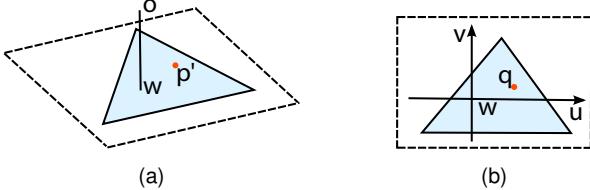


Fig. 3. Mapping a 3D triangle (a) into 2D (b) by parameterizing the triangle in the 2D space defined by the plane of the triangle. The origin \mathbf{o} of the 3D space is projected onto the plane of the triangle at \mathbf{w} that is treated as the origin of the 2D space. 2D equivalent of each point \mathbf{p}' within the triangle is denoted as \mathbf{q} .

method can generate more accurate results compared with KDE over discrete sample points. Also, by computing SDE analytically instead of performing KDE on a large number of sample points, the computation cost reduces. Our method transforms the calculation of SDEs into the calculation of integrals of the standard bivariate normal distribution over 2D triangular areas by parameterizing each triangular patch in 2D space as explained below.

4.2.1 Transforming Surface Density into Bivariate Normal Integral

The surface density for a triangle T is defined as a surface integral of the kernel $K_{\mathbf{H}}$ over T , as shown in Equation 5. By parameterizing the triangle in 2D space, this surface integral can be transformed into an ordinary integral over a 2D triangular area. To achieve this, we first transform T based on the target position \mathbf{x} and the bandwidth matrix \mathbf{H} by applying the transformation

$$\mathbf{p}' = \mathbf{H}^{-\frac{1}{2}}(\mathbf{x} - \mathbf{p}) \quad (7)$$

for each vertex $\mathbf{p} \in T$. After the transformation, the triangle T is transformed into a new triangle T' , and the surface density of T becomes an integral of the kernel function K over the triangle T' ,

$$\tau_{\mathbf{H}}(\mathbf{x}, T) = c_T \iint_{T'} K(\mathbf{p}') dT', \quad (8)$$

where $c_T = |\mathbf{H}|^{-\frac{1}{2}} \frac{d|T|}{dT'} = |\mathbf{H}|^{-\frac{1}{2}} \frac{\text{Area}(T)}{\text{Area}(T')}$. After the transformation, the surface density of T depends only on the kernel function K and the transformed triangle T' .

We now parametrize the triangle T' in the 2D space and transform the surface integral over the triangle T' into the integral over a 2D triangular area. We first project the 3D origin \mathbf{o} onto the plane of the triangle T' and use the projected point \mathbf{w} as the origin of the 2D space, as shown in Figure 3 (a). 2D equivalent of each point $\mathbf{p}' \in T'$ is then defined as $\mathbf{q} = (u, v)^T$ in \mathbb{R}^2 with w as the origin, as shown in Figure 3 (b). Because the standard multivariate normal kernel, which is rotation invariant, is used as the kernel function K , the integral of K over the triangle T' is transformed to the integral over a 2D region \bar{T} :

$$\tau_{\mathbf{H}}(\mathbf{x}, T) = c_T \iint_{\bar{T}} K((u, v, \lambda)^T) du dv, \quad (9)$$

where $\lambda = \|\mathbf{w} - \mathbf{o}\|$. The standard multivariate normal kernel K is further decomposed into the product of lower dimensional standard normal distributions, and the integral of K over the 2D region \bar{T} can be defined as a product of the

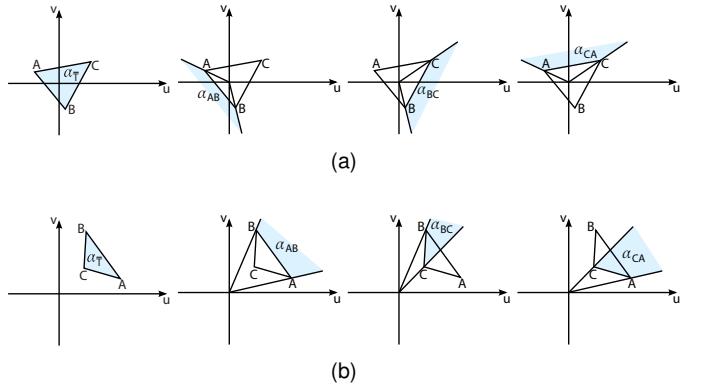


Fig. 4. Computing the bivariate normal integral over a triangle \bar{T} , denoted as $\alpha_{\bar{T}}$. Integrals α_{AB} , α_{BC} , and α_{CA} are first computed based on the method proposed by Owen [58]. Then $\alpha_{\bar{T}}$ can be computed by combining α_{AB} , α_{BC} , and α_{CA} , for which there are two cases. (a) The origin is inside the triangle and (b) The origin is outside the triangle.

density of a 1D standard normal distribution at position λ and an integral of the 2D standard normal distribution over the 2D region \bar{T} ,

$$\tau_{\mathbf{H}}(\mathbf{x}, T) = c_T N(\lambda) \iint_{\bar{T}} N_2((u, v)^T) du dv, \quad (10)$$

where $N(\lambda)$ and $N_2((u, v)^T)$ are the standard uni- and bivariate normal distributions, respectively.

4.2.2 Computing Bivariate Normal Integral

To calculate the bivariate normal integral over \bar{T} , denoted as $\alpha_{\bar{T}}$, we use the method proposed by Owen [58] and extended by Donnelly [59]. Let O be the origin and AB be an edge of a given triangle, Owen's method gives the bivariate normal integral over an area defined by line OA , OB , and beyond line AB , denoted as α_{AB} . (Details in Appendix A.)

To calculate the integral of the standard bivariate normal distribution over a triangle \bar{T} defined by vertices A , B , and C , we first calculate integrals α_{AB} , α_{BC} , and α_{CA} , and then combine them to get the integral $\alpha_{\bar{T}}$. There are two cases for combining α_{AB} , α_{BC} , and α_{CA} , as illustrated in Figure 4. In Figure 4 (a), the origin is inside the triangle, and the integral $\alpha_{\bar{T}}$ is computed with

$$\alpha_{\bar{T}} = 1 - \alpha_{AB} - \alpha_{BC} - \alpha_{CA}. \quad (11)$$

If the origin is outside the triangle, as shown in Figure 4 (b), the integral $\alpha_{\bar{T}}$ is computed with

$$\alpha_{\bar{T}} = \alpha_{BC} + \alpha_{CA} - \alpha_{AB}. \quad (12)$$

5 BANDWIDTH SELECTION WITH LEAST SQUARES CROSS VALIDATION (LSCV)

We propose a method to guide the selection of the bandwidth based on the variability of ensemble surfaces. The crucial parameter for computing SDEs is the bandwidth matrix \mathbf{H} , because it controls the degree and orientation of smoothing induced [60]. There are three classes of the bandwidth matrix \mathbf{H} : positive scalars times the identity matrix $\mathbf{H} = \sigma^2 \mathbf{I}$, which have the same amount of smoothing in all directions; diagonal matrices $\mathbf{H} = \text{diag}(\sigma_0^2, \sigma_1^2, \dots, \sigma_{d-1}^2)$,

with individual smoothing in each of the dimensions; and symmetric positive definite matrices, which allow arbitrary amount and orientation of smoothing. In this work, we use the first class of the bandwidth matrix $\mathbf{H} = \sigma^2 \mathbf{I}$, which assumes a single smoothing for all directions that is controlled by a positive constant value σ^2 . Note that σ may result in different scales in the physical space, if the grid coordinate system scales differently on each dimension.

To select a proper σ , we use the most common optimality criterion, which is the mean integrated squared error (MISE) between the SDE $\hat{\mathcal{S}}_{\mathbf{H}}(\mathbf{x})$ with respect to a bandwidth matrix \mathbf{H} and the target probability density function $\mathcal{S}(\mathbf{x})$:

$$\text{MISE}(\mathbf{H}) = E \left[\int_{\mathbb{R}^3} (\hat{\mathcal{S}}_{\mathbf{H}}(\mathbf{x}) - \mathcal{S}(\mathbf{x}))^2 d\mathbf{x} \right]. \quad (13)$$

Because the target probability density function $\mathcal{S}(\mathbf{x})$ is unknown, we use the LSCV method [61], [62] to approximate the MISE. The existing LSCV method is defined for a set of discretized data points $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{n-1}$, which approximates the MISE as

$$\text{LSCV}(\mathbf{H}) = \int_{\mathbb{R}^3} \hat{f}_{\mathbf{H}}(\mathbf{x})^2 d\mathbf{x} - \frac{2}{n} \sum_{i=0}^{n-1} \hat{f}_{-i}(\mathbf{x}_i), \quad (14)$$

where $\hat{f}_{\mathbf{H}}(\mathbf{x})$ is the KDE at position \mathbf{x} , and $\hat{f}_{-i}(\mathbf{x}_i)$ is the leave-one-out estimator that estimates the density at position \mathbf{x}_i for the input data points except \mathbf{x}_i :

$$\hat{f}_{-i}(\mathbf{x}_i) = \frac{1}{n-1} \sum_{j=0, j \neq i}^{n-1} K_{\mathbf{H}}(\mathbf{x}_i - \mathbf{x}_j). \quad (15)$$

We extend the LSCV for ensemble surfaces by redefining the leave-one-out estimator. For an ensemble of surfaces S_0, S_1, \dots, S_{N-1} , we subsequently take out one surface S_i , then estimate the surface density for the remaining surfaces. Like LSCV, we take out one surface S_i each time, then compute the leave-one-out estimator on points sampled from the surface S_i and accumulate the results. The LSCV for ensemble surfaces is then defined as

$$\text{LSCV}(\mathbf{H}) = \int_{\mathbb{R}^3} \hat{\mathcal{S}}_{\mathbf{H}}(\mathbf{x})^2 d\mathbf{x} - \frac{2}{A(S)} \sum_{i=0}^{N-1} \iint_{S_i} \hat{\mathcal{S}}_{-i}(\mathbf{p}) dS_i. \quad (16)$$

To pick a proper bandwidth matrix based on the smoothing factor σ^2 , we minimize the function $\text{LSCV}(\mathbf{H})$ for σ values within the interval $(0, +\infty)$. To this end, we use the golden-section search method [63], which successively narrows the interval of σ in which the minimum of $\text{LSCV}(\mathbf{H})$ occurs until the interval is small enough. In this work, we start from an interval $(0, a]$, which satisfies a condition that within $(0, a]$ we can find at least one σ that gives a $\text{LSCV}(\mathbf{H})$ smaller than the $\text{LSCV}(\mathbf{H})$ given by a .

When the ensemble surfaces have similar shape and spatial location, the leave-one-out estimator decreases more slowly than $\int \hat{\mathcal{S}}_{\mathbf{H}}(\mathbf{x})^2 d\mathbf{x}$ increases when the smoothing factor σ^2 of the bandwidth matrix decreases, because the other surfaces are close to the removed surface. On the other hand, if the ensemble surfaces are far away from each other, the leave-one-out estimator decreases rapidly when the smoothing factor σ^2 of the bandwidth matrix decreases. Because there is no analytic solution for the evaluation of

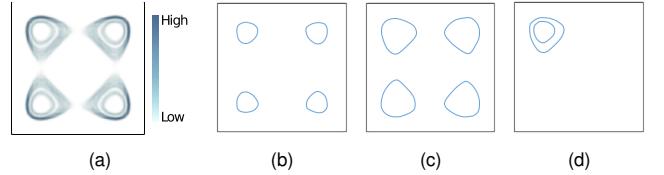


Fig. 5. (a) Density field of an ensemble of isocontours in 2D. (b)-(d) Three possible shapes of the isocontours.

$\text{LSCV}(\mathbf{H})$, we evaluate it numerically based on Riemann sums in this work.

6 VISUAL EXPLORATION OF SDE

We design a visual exploration framework called *eFESTA* to visualize and analyze ensemble surfaces based on SDE. Based on the method presented in preceding sections, an ensemble of surfaces S_0, S_1, \dots, S_{N-1} can be transformed into a SDE, which represents the spatial distribution of the surfaces. The SDE is represented as a scalar field where the scalar value at a given grid point represents the probability density of the surfaces passing through it. By visualizing the SDE using existing volume exploration techniques such as volume rendering, the user is able to get an overview of how the surfaces are distributed in the spatial domain. However, the shape and the major trends of the surfaces require further analysis and exploration. For example, in Figure 5 (a) the density estimation result for an ensemble of isocontours in 2D is visualized. Based on the visualization, it is difficult to determine the shape of the isocontours. Various possible shapes of the isocontours can be inferred from the visualization of the SDE, such as the three possible shapes shown in Figures 5 (b), (c), and (d). It is also difficult to get the percentage of a particular class of shape. To answer these questions, our framework transforms the surfaces S_0, S_1, \dots, S_{N-1} to an ensemble of density fields D_0, D_1, \dots, D_{N-1} based on their contribution to the SDE. In other words, the SDE equals to the sum of D_0, D_1, \dots, D_{N-1} . We then extend the method presented in [64] to organize the density fields into a hierarchical representation based on the similarity between them. The resulting hierarchical representation is used to guide visual exploration of the density fields as well as their corresponding surfaces. Below we describe the method to organize density fields hierarchically and the design considerations and choices of our interface.

6.1 Hierarchical Representation of Density Fields

For efficient visual exploration, we organize density fields into a hierarchical representation based on the similarity between them. We first compute a distance matrix containing the distances between every pair of the density fields and organize the density fields into a binary tree based on hierarchical clustering. Below are the key steps of constructing the hierarchical representation.

We treat each density field as a point in high-dimensional space \mathbb{R}^m , where m equals to the number of grid points in the density field. Then the distance between the density fields is computed as the l^2 -norm distance in \mathbb{R}^m , and stored in a distance matrix. Based on the distance matrix, agglomerative hierarchical clustering is then performed to build a hierarchy of clusters (an unbalanced binary tree) of

TABLE 1. Data specifications and performance (in seconds). t_d : timings for density estimation; t_c : timings for hierarchical clustering; t_r : response time for interactive queries (i.e., selecting a node in the hierarchical clustering tree).

| Dataset | Application | Number of Members | Number of Triangles | Resolution of Density Field | Bandwidth σ | Performance (s) | | |
|-----------------|---------------|-------------------|---------------------|-----------------------------|--------------------|-------------------|-------|--|
| | | | | | | t_d | t_c | t_r |
| Tangle | Isosurface | 50 | 330,772 | 249 × 249 × 249 | 0.185 | 1.3×10^2 | 1.78 | $3.6 \times 10^{-3} \sim 4.8 \times 10^{-2}$ |
| MBST-98 | Isosurface | 600 | 307,044 | 179 × 105 × 31 | 0.38 | 0.9×10^1 | 2.26 | $8.4 \times 10^{-4} \sim 6.1 \times 10^{-2}$ |
| HRRR | LCS | 100 | 4,773,666 | 224 × 132 × 36 | 0.67 | 1.9×10^2 | 1.37 | $1.3 \times 10^{-3} \sim 2.7 \times 10^{-2}$ |
| Square Cylinder | Streamsurface | 40 | 43,636 | 383 × 127 × 95 | 0.3 | 0.8×10^1 | 0.82 | $2.2 \times 10^{-3} \sim 1.3 \times 10^{-2}$ |

the density fields in a bottom-up manner. Starting with each density field in a separate cluster, the hierarchy is built by merging pairs of similar clusters until all density fields are contained in a single cluster. Distance between clusters is determined based on Ward's method [65], which calculates the distance between a pair of clusters as the change of the total sum of squares after merging. In this way, the sum of the distances between data points and the new cluster center after merging is minimized. After clustering, similar density fields are grouped into the same subtree, from which the major trends of the surfaces can be extracted and visualized by selecting subtrees.

6.2 Visual Exploration with Hierarchical Clustering Tree

We design a system for interactive visual exploration of density fields and their corresponding surfaces guided by the hierarchical clustering tree generated before. Our system consists of two linked views: a hierarchical clustering tree view and a spatial view.

The hierarchical clustering tree view is provided to help users select and explore ensemble surfaces at different hierarchies. In this work, a radial tree layout is chosen to visualize the hierarchical clustering tree. Compared with a linear tree layout, the radial tree layout utilizes space more effectively. The root of the hierarchical clustering tree is placed at the center of the layout, and the distance of a node from the root encodes the level of the node. A ring layout is then drawn around the hierarchical clustering tree to indicate the surface features extracted from each ensemble member. The surface features are ordered based on their IDs in the ring layout. When the user selects a node of interest, the corresponding ring segments are simultaneously enlarged and highlighted. The user can also adjust the total number of leaf nodes (i.e., the maximum number of clusters in the hierarchical clustering) in the tree layout to reduce visual clutter. In this work, the total number of leaf nodes is set to 20. Figure 6 shows the visualization of an example hierarchical clustering tree view.

The spatial view has two subviews: a density field view and a surface view. The density field view is used to visualize a density field of interest, and the surface view shows the corresponding surfaces. After the user selects one

node in the hierarchical clustering tree view, a density field is constructed on the fly by aggregating the density fields in the leaf nodes of the selected subtree; the density field view is updated interactively with direct volume rendering. We color the density fields using a single-hue colormap from dark green to light green, and the user can adjust the opacity transfer function to highlight regions with different density values. The surfaces of corresponding ensemble members are shown in the surface view as spaghetti plots. The surfaces are rendered with transparency such that the interior of the surfaces can be visualized. The spatial view is linked with the hierarchical clustering tree view. When the user hovers over a node of interest, the spatial view is simultaneously updated to visualize the density field and the surfaces of the selected ensemble members.

7 IMPLEMENTATION DETAILS AND PERFORMANCE EVALUATION

We parallelize the calculation of SDE over grid points using a GPU to handle millions of triangular patches over a density field discretized by millions of grid points. In our implementation, each GPU thread estimates the surface density at a grid point by iterating over all the triangles of the given surface. To further improve the performance of the SDE calculation, we filter out triangles whose contribution to the density at a target grid point is negligible. For each triangle, we first construct its axis-aligned bounding box and enlarge the bounding box for 4σ , where σ is the bandwidth. The reason for using 4σ is because the probability of a Gaussian distribution over the region within 4σ is greater than 99.99%, and we hence neglect the contribution of a Gaussian distribution to the density at a location whose distance to the mean is greater than 4σ . For a given grid point, we test whether it is within the bounding box or not. If the grid point is within, we evaluate the surface density of the triangle over the grid point; if not, then we consider the triangle has zero contribution to the density at the grid point so no density estimation for that triangle is performed. The resulting density fields are stored as textures in the GPU memory for further visualization and exploration.

The prototype system is implemented with C++ and CUDA. All the experiments were performed on a workstation with two Intel Xeon E5620 CPUs (2.40 GHz), 12 GB memory, and an Nvidia Tesla K40c GPU. Table 1 reports the timings for the proposed ensemble exploration framework. The density estimation takes 1 to 3 minutes, and the hierarchical clustering takes a couple of seconds. The response time after the user selecting a node in the hierarchical clustering tree is less than 0.1 second.

8 RESULTS

In this section, we first compare SDE with two alternative

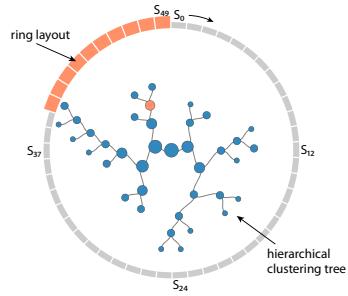


Fig. 6. Hierarchical clustering tree view, where ring layout represents surfaces ordered by their IDs from S_0 to S_{N-1} . When a node is selected, the corresponding ring segments are enlarged and highlighted.

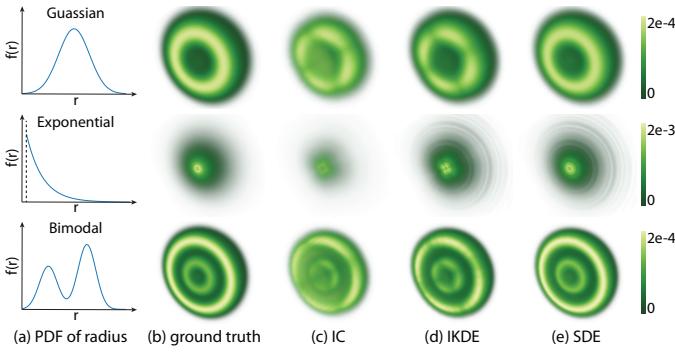


Fig. 7. Visualization of the three synthetic ensembles of spheres: (a) distributions of radius; (b) ground truth density fields; (c)-(e) density estimation results of IC, IKDE, and our method, respectively.

approaches using a synthetic ensemble of surfaces that have known spatial distributions in Section 8.1. The comparison shows that the proposed method generates more accurate density estimation results than the other two approaches when the computation costs are similar. Then we show the usefulness and generality of SDE by applying it on ensemble surfaces for three applications, including ensemble isosurfaces in Section 8.2, LCSs in uncertain unsteady flows in Section 8.3, and streamsurfaces in ensemble fluid flows in Section 8.4. For all three applications, the proposed method is able to characterize how the surfaces are distributed in the spatial domain by only using polygon meshes. Moreover, for the application of ensemble isosurfaces, the proposed method is able to generate comparable results to previous methods that require scalar fields to be available.

8.1 Synthetic Ensemble Surfaces

In this experiment, we focused on evaluating the accuracy of the proposed density estimation method and comparing it with alternative approaches. To this end, we first synthesized ensemble surfaces with respect to a predefined spatial distribution. Density estimation approaches were then performed on the generated surfaces. Finally, the resulting density fields were compared with the spatial distribution that was used to guide the generation of the surfaces.

The ensemble surfaces used for evaluation were based on a sphere centered at the origin with an uncertain radius. The uncertainty of the radius was regarded as a random variable R that was defined as a PDF $f(r)$. Three different $f(r)$ were tested in the experiment, including a Gaussian distribution, a shifted exponential distribution, and a bimodal distribution (i.e., a Gaussian mixture modeled with two Gaussian components) as shown in Figure 7(a). Based on $f(r)$, the probability density of a spatial location $\mathbf{x} \in \mathbb{R}^3$ on the sphere can be computed analytically as follows. We first compute the distance d from \mathbf{x} to the center of the sphere and then evaluate $f(r)$ at d . In the end, we normalize the result such that the density values integrate to 1 for all $\mathbf{x} \in \mathbb{R}^3$. In the experiment, the density values were evaluated on the grid points of a regular grid with a spatial resolution of 256^3 . The resulting density field was treated as the ground truth, as shown in Figure 7(b) using volume rendering. A clipping plane is placed at the center of the sphere to make the interior visible. An ensemble of spheres was then generated based on each $f(r)$. A set of radiiuses

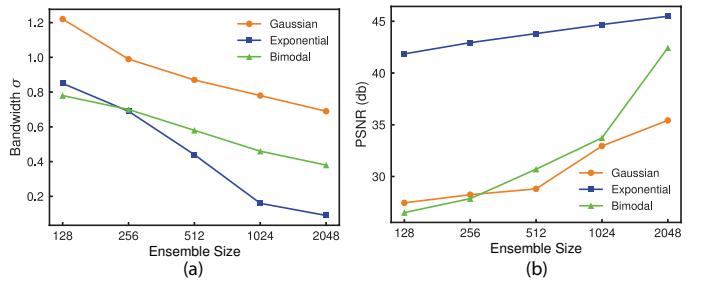


Fig. 8. Quantitative analysis of the proposed method for the three synthetic ensembles of spheres with different ensemble sizes from 128 to 2,048 for every power of 2: (a) bandwidth σ suggested by the proposed bandwidth selection method, (b) PSNR (db) between the density estimation results of the proposed method and the ground truth.

was first sampled randomly based on $f(r)$, and then a sphere mesh was constructed for each radius by subdividing the sphere along θ and ϕ in spherical coordinates.

The SDE and two alternative density estimation approaches were applied on the ensemble of sphere meshes, which are detailed as follows.

SDE The proposed bandwidth selection method was first applied on the ensemble of sphere meshes. Then the SDE was calculated based on the selected bandwidth.

Intersection count (IC) First, a regular grid was defined over the spheres. Then, the count of sphere meshes intersecting each grid cell was computed based on triangle-box intersection [66]. Finally, the count of each cell was normalized by the total count to estimate the density field.

Intersection-based KDE (IKDE) Similar to IC, a regular grid was defined over the sphere meshes first. Then, for each sphere mesh, we extracted active cells (i.e., grid cells that intersect the sphere meshes). Finally, we performed point-based KDE over the center of the active cells to estimate the density field. For comparison, the bandwidth used for KDE was the same as SDE.

The resulting density fields were then compared with the ground truth to evaluate the accuracy and performance of the three approaches.

We first evaluated the accuracy of the proposed method by applying it to ensembles of spheres with different ensemble sizes (i.e., number of ensemble members) from 128 to 2,048 for every power of 2. The SDEs were computed on a regular grid with a spatial resolution of 256^3 . The bandwidth selection results with respect to different ensemble sizes are shown in Figure 8(a), we can see that more ensemble members lead to a smaller bandwidth. Figure 8(b) shows the peak signal-to-noise ratio (PSNR) between the SDE and the ground truth. We can see that the PSNR increases when the ensemble size increases, because more samples can better fit the underlying distribution. Also, our method can generate density fields that are close to the ground truth. For example, the PSNRs are greater than 30db for all three $f(r)$ when the ensemble size is greater than 1,024.

We then compared the accuracy and performance of the proposed method with IC and IKDE. Because the grid resolution plays an important role in the accuracy of IC and IKDE, we compared the three approaches on different grid resolutions from 8^3 to 256^3 for every power of 2. In order to compare with the ground truth, the resulting density fields were then upsampled to 256^3 using trilinear interpolation.

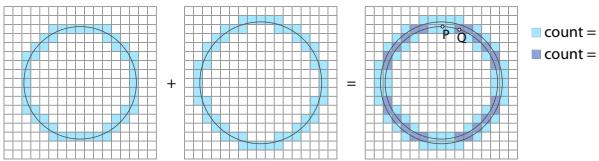


Fig. 9. Example of counting the number of circles intersecting grid cells. Because of the discretization, errors are introduced into the resulting density field. For example, points P and Q should have the same density, but different densities are assigned to P and Q in the result.

The ensemble size is fixed to 2,048 in this comparison. For all three approaches, the computation was parallelized over grid points using GPU.

The visualization of the resulting density fields generated by IC, IKDE, and our method are shown in Figures 7(c), (d), and (e), respectively. The grid resolution of each approach in Figure 7 is the resolution that corresponds to the best density estimation result, which are 32^3 , 256^3 , and 256^3 for IC, IKDE, and our method, respectively. We can see that the density fields generated by our method are visually similar to the ground truth, and undesirable artifacts are introduced into the density fields generated by IC and IKDE. This is because after discretizing surface patches with respect to a given grid, the information of the surface patches (e.g., location, orientation, and shape) within each grid cell is lost, which will introduce discretization error into the density estimation results. Figure 9 shows a 2D example of such discretization error. In this example, a density field is estimated for two circles using IC. We can see that different density values are assigned to spatial locations that should have the same density value (e.g., points P and Q). This discrepancy is caused by the discretization of space. Although increasing the grid resolution can reduce the discretization error, it will reduce the accuracy and performance of IC and IKDE, which are discussed in detail as follows.

The PSNRs between the density fields and the ground truth are shown in Figures 10 (a), (c), and (e). Also, the performance of the three approaches are shown in Figures 10 (b), (d), and (f). We can see that IC takes the least amount of time, but generates the worst results for all three $f(r)$. Also, the accuracy of IC increases in the beginning and starts decreasing when the grid resolution is greater than 32^3 . This is because on one hand, the discretization error decreases as the grid resolution increases, and on the other hand, the size of grid cells is too small to capture the spatial distribution of the surfaces when the grid resolution is too high. More specifically, if the size of grid cells is too small, most grid cells only intersect individual surface patches once. As a result, the density field becomes sparse and noisy. IKDE does not suffer from this problem when the grid resolution is too high, because even most grid cells only intersect individual surface patch, the intersected cells can contribute density to neighboring cells by using KDE. Hence, increasing grid resolution can improve the accuracy of IKDE, as shown in Figures 10(a), (c), and (e). However, when the grid resolution increases, the number of active cells increases as well. As a result, IKDE needs to perform KDE on more points. In contrast, our method performs density estimation on the same number of triangular patches when the grid resolution changes. Hence, the computation cost of IKDE increases faster than our method when the grid resolution

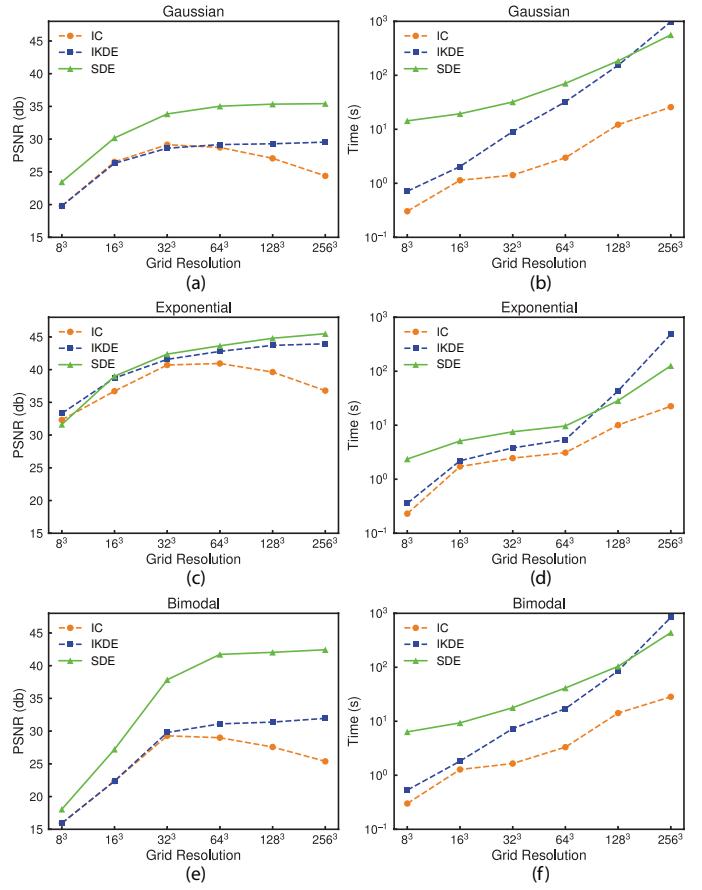


Fig. 10. Quantitative comparison for IC, IKDE, and our method: (a), (c), and (e) PSNR (db) between the density estimation results and the ground truth; (b), (d), (f) timings for density estimation.

increases, as shown in Figures 10(b), (d), and (f). When the grid resolution equals to 256^3 , our method takes less time than IKDE and produces more accurate results.

8.2 Isosurfaces in Ensemble Scalar Fields

We applied the proposed method on two ensemble scalar fields: a synthetic ensemble dataset based on the tangle function [67] and a real-world ensemble simulation data in environmental science. For the synthetic dataset, we compared the proposed method with the state-of-the-art approaches on uncertain isosurface visualization.

8.2.1 Synthetic Ensemble Scalar Fields

Our first experiment was performed on a synthetic ensemble dataset based on the tangle function. The ensemble was obtained by injecting multiple realizations of noise in the scalar field. The noise samples were generated by sampling 10 Gaussian kernels with shifted means and a fixed standard deviation 0.18, with 5 samples per kernel, to create an ensemble scalar dataset with 50 members. Seven kernels were placed closer to the underlying value, and three kernels were placed farther away. An ensemble of isosurfaces was then extracted for the isovalue of -0.6 . Figure 11 (a) shows the spaghetti plots of the isosurfaces with one slice extracted and highlighted. The slice is defined by a plane at the location $(0, 0, 21.75)$ with a normal $(0, 0, 1)$.

When scalar fields are available, various techniques can be used to model the positional uncertainty of isosurfaces. In

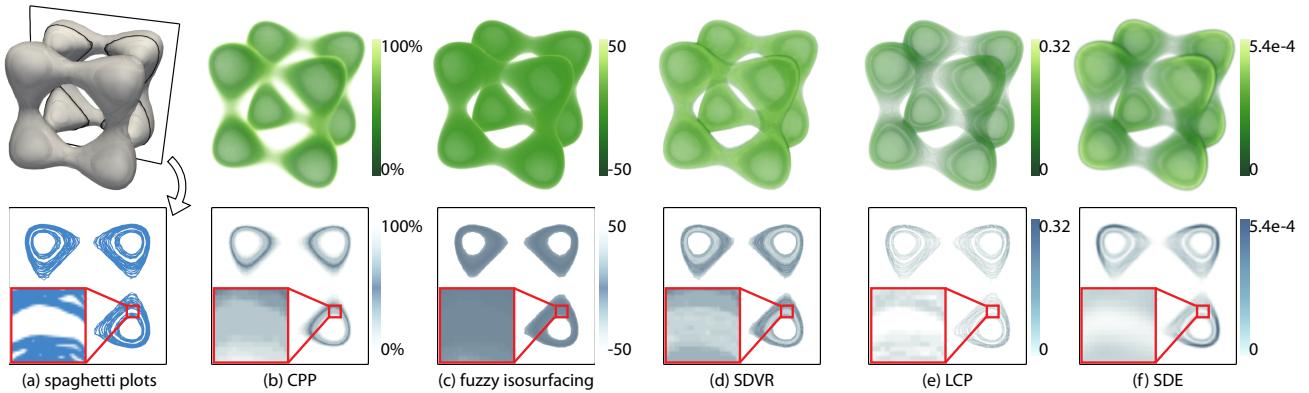


Fig. 11. Visualization of the isosurfaces extracted from the synthetic ensemble scalar fields using different methods: (a) spaghetti plots, (b) CPP [9], (c) fuzzy isosurfacing [1], (d) SDVR [8], (e) LCP [31], and (f) SDE. Notice that different techniques quantify the positional uncertainty of isosurfaces with different metrics. Hence, the results have different value ranges and have to be visualized using specific transfer functions.

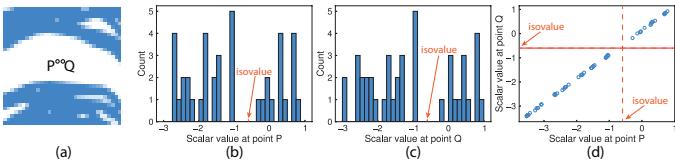


Fig. 12. Example of correlation between random variables at different spatial locations: (a) locations P and Q that random variables are defined on, (b) histogram of scalar values at P, (c) histogram of scalar values at Q, and (d) scatter plot of scalar values at P and Q.

this experiment, we compared our method with four techniques that are working on uncertain scalar fields, including: contour probability plots (CPPs) [9], fuzzy isosurfacing [1], statistical direct volume rendering (SDVR) [8], and level crossing probabilities (LCPs) [31] detailed as follows.

CPP For each grid point, the CPP is defined as the percentage of scalar values that are greater than the isovalue, which is from 0% to 100%. Given an ensemble of scalar fields, the number of ensemble members that have scalar values greater than the isovalue is first computed at each grid point. Then, the counts are normalized by the total number of ensemble members to get CPPs. Notice that locations that are considered having a higher chance to be on the isosurface are the locations with CPPs around 50%. Hence, a triangle function centered at 50% is often used as the transfer function to render a CPP field.

Fuzzy Isosurfacing Similar to CPP, fuzzy isosurfacing first computes the number of scalar values above and below the isovalue at each grid point, denoted as a and b , respectively. The likelihood of the presence of the isosurface is then computed for each grid point as $g = \frac{a}{b} - \frac{b}{a}$. If $a = b$, g is set to zero; if $a = 0$ or $b = 0$, g is set to $-N$ and N , respectively, where N is the number of ensemble members. Because we have 50 ensemble members in this experiment, the value of g is between -50 and 50. Fuzzy isosurfacing is typically visualized using volume rendering with a transfer function defined by triangle function centered at 0, because locations that are considered having a higher chance to be on the isosurface are the locations with $a = b$.

SDVR In SDVR, the distribution of scalar values at each grid point is modeled as box-splines, which enables interpolation of distributions at arbitrary location along the viewing rays. The transfer function is then integrated against the interpolated distribution at each sample point

along the viewing rays to obtain the expected color and opacity. The expected colors and opacities are then blended to get the volume rendering results. To visualize uncertain isosurface of a give isovalue using SDVR, a triangle function centered at the isovalue is used as the transfer function.

LCP Unlike the aforementioned methods, the LCP is defined on grid cells instead of grid points. The LCP of a grid cell is defined as the probability of the isosurface intersecting the cell, which is from 0 to 1. To get the LCP of a grid cell defined by several grid points, the joint distribution is first constructed for the random variables defined on the grid points. The LCP is then approximated by sampling from the joint distribution and computing the percentage of samples that the isosurface intersecting the cell.

SDE In contrast with previous methods, SDE is computed based on the geometries of the isosurfaces. First, the proposed bandwidth selection method is applied on the isosurfaces, which suggests to use a bandwidth equals 0.185. Then, SDE is computed based on the selected bandwidth. Because SDE represents the PDF of the spatial distribution of the isosurfaces, SDE integrates to 1 in the spatial domain.

The volume rendering results of the CPP, Fuzzy Isosurfacing, SDVR, LCP, and SDE are shown in Figure 11(b)-(f), respectively. Compared with the isosurfaces shown in Figure 11(a), we can see that the results of CPP, Fuzzy Isosurfacing, and SDVR assign nonzero probabilities to locations that no isosurfaces are passing through (e.g., empty regions enlarged in Figure 11(a)). This is because the three techniques model the uncertainty of the scalar values at each grid point as an independent random variable without considering correlations between the random variables at different spatial locations. For example, for the adjacent points P and Q selected from the empty regions in Figure 12(a), the distributions of the scalar values at points P and Q are shown in Figure 12(b) and (c), respectively. If we model the uncertainty of the scalar values at the two points as independent random variables. The probability that the isosurface passing through any points between P and Q is nonzero, because it is possible that the scalar value at P is greater than the isovalue and the scalar value at Q is less than the isovalue. However, the random variables at P and Q are correlated. Figure 12(d) shows the scatter plots of the scalar values at P and Q. We can see that when the scalar value at P is greater than the isovalue, the scalar value at

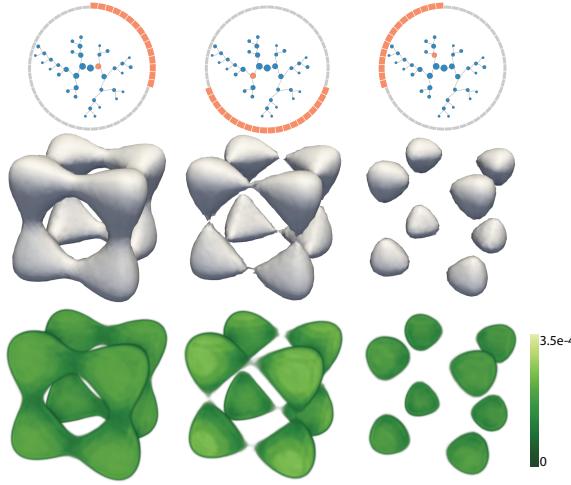


Fig. 13. Visualization of the density fields and the underlying isosurfaces for three representative clusters selected from the hierarchical clustering tree for the synthetic ensemble scalar fields. The major trends of the isosurfaces are extracted and highlighted.

Q is also greater than the isovalue, and vice versa. Hence, the isosurface does pass through any points between P and Q . By considering the correlations of the random variables within each grid cell, the result of LCP model the positional uncertainty of isosurface more correctly. Because the SDE is computed based on isosurfaces that are extracted from each ensemble member independently, spatial correlations of scalar values within each ensemble member are preserved. Hence, the result of SDE is comparable to the result of LCP. However, the previous techniques require uncertain scalar fields as input to model the positional uncertainty of isosurfaces. For applications such as LCSs and streamsurfaces, no field data can be directly used to model the positional uncertainty of the surfaces. Unlike the previous techniques, our method is able to model the positional uncertainty of derived surfaces features, which can be applied on LCSs and streamsurfaces as discussed in the following sections 8.3 and 8.4. Moreover, unlike the previous techniques which only produce one final result for uncertain isosurfaces, our approach organizes ensemble isosurfaces into a hierarchical representation based on their similarities and enables interactive visual exploration of the major trends. Figure 13 shows the results of the proposed visual exploration method. By hierarchical clustering, the isosurfaces are clustered based on the similarities between their contribution to the SDE, and a hierarchical clustering tree is visualized to guide the exploration of the isosurfaces. Three representative clusters of isosurfaces are shown in Figure 13, with each column representing one cluster. The contribution of each cluster to the SDE is represented as a density field and visualized using volume rendering. Compared with previous methods, the proposed method is able to extract and highlight the major trends of the isosurfaces.

8.2.2 Real-world Ensemble Simulation Data

Our second experiment was performed on the Massachusetts Bay Sea Trial (MBST-98) ensemble dataset, which is an interdisciplinary (i.e., physical—biogeochemical) forecast simulation based on the Littoral Ocean Observing and Predicting System [68], [69]. The simulation generated an

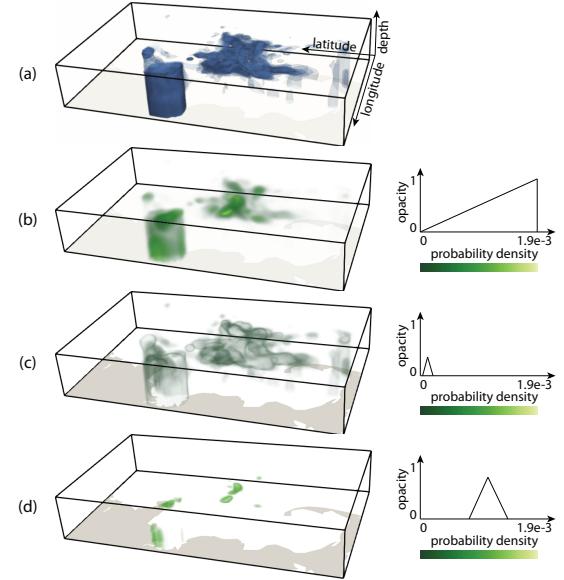


Fig. 14. Visualization of the isosurfaces extracted from the CHL variable of the MBST-98 ensemble data with an isovalue 1.4 using (a) spaghetti plots and (b)-(d) SDE with different transfer functions. The vertical axis represents water depth in meters, which increases from bottom to top.

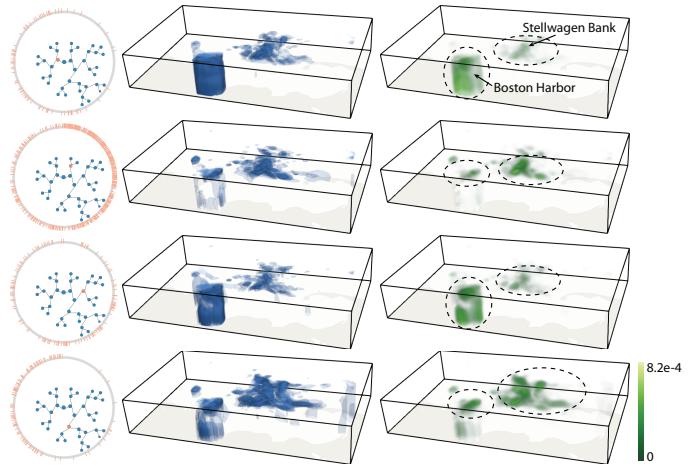


Fig. 15. Visualization of the density fields for four representative clusters based on the isosurfaces generated from the CHL variable of the MBST-98 ensemble data with an isovalue 1.4. Distinct features are highlighted at Boston Harbor and Stellwagen Bank for the four clusters.

ensemble of 600 members with more than 30 variables from August 17 to October 5, 1998 in Massachusetts Bay. In this experiment, we select the CHL variable, which represents the chlorophyll-a concentration, at the time step representing September 2, 1998. We extracted the isosurface at isovalue 1.4 mg/m^3 for each ensemble member, because phytoplankton are contaminated in regions with CHL equals 1.4 mg/m^3 . The SDE is then computed for the extracted isosurfaces. The isosurfaces are visualized using spaghetti plots as shown in 14 (a). By applying the proposed bandwidth selection method, the bandwidth σ was set to 0.38. The SDE is visualized using volume rendering with three different transfer functions as shown in Figure 14 (b)-(d). A linear opacity transfer function is used in Figure 14(b). By comparing with the spaghetti plots, we can see that the SDE can reveal the distribution of the surfaces. Moreover, by adjusting the transfer function, the low and medium probability density regions can be highlighted in Figure 14

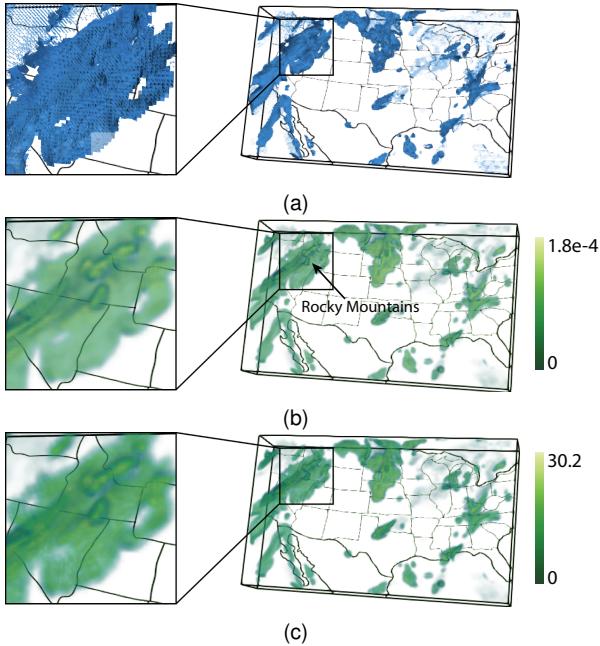


Fig. 16. Visualization of the uncertain LCSs extracted from the HRRR ensemble dataset: (a) spaghetti plots of the surfaces, (b) SDE, and (c) density field created by SPH.

(c)-(d). We can see that the low probability density regions are corresponding to the outline surfaces.

Figure 15 visualizes four selected clusters of isosurfaces based on their contribution to the SDE, where each row represents one cluster. The contribution of each cluster to the SDE is represented as a density field and visualized using volume rendering. Overall, the isosurfaces with chlorophyll-a concentration 1.4 mg/m^3 have a higher likelihood to be located at Boston Harbor and Stellwagen Bank compared with other regions. By comparing the density fields across clusters, distinct features are highlighted. For example, the density field in the second cluster reveals that the isosurfaces have lower likelihood at Boston Harbor compared with other clusters; and density fields in the first, third, and fourth clusters highlight different regions at Boston Harbor. Many other features can also be identified by exploring the density fields guided by the hierarchical clustering tree.

8.3 Lagrangian Coherent Structures in Uncertain Flow Fields

We also used the proposed technique for visualizing and analyzing uncertain LCSs [10] for uncertain unsteady flow fields. The LCSs are surfaces that represent the boundaries between attracting or repelling particles in unsteady flow fields. Because the positional uncertainty of LCSs can not be derived from the flow fields directly, Guo et al. [40] presented a technique to visualize and analyze uncertain LCSs by extracting an ensemble of LCSs and then constructing a density field through a density estimation method based on SPH and zero-order kernel functions. They calculated the density of an arbitrary location \mathbf{x} by calculating the area of the surfaces within a sphere centered on \mathbf{x} , and then normalized the results by the volume of the sphere.

In this experiment, we calculated SDE for the uncertain LCSs extracted from High-Resolution Rapid Refresh (HRRR) data [70] and compared with SPH. The HRRR is

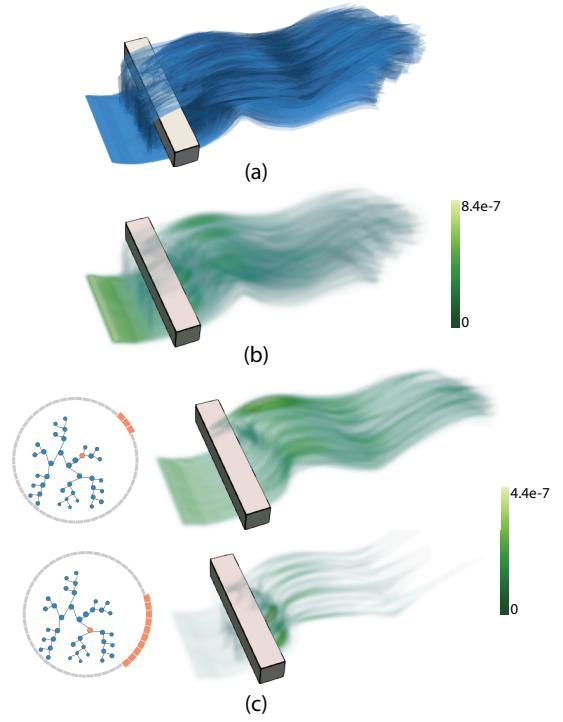


Fig. 17. Visualization of the streamsurfaces generated from the synthetic ensemble flow around a confined square cylinder, where the flow is moving from left to right: (a) spaghetti plots of the streamsurfaces (b) volume rendering of the SDE, and (c) two clusters of density fields that highlight different flow features.

a National Oceanic and Atmospheric Administration real-time atmospheric model. The HRRR is updated hourly and produces a forecast up to 16 hours. It is available to the public through the Unidata's THREDDS Data Server 2. We used an ensemble of the HRRR simulation outputs to extract uncertain LCSs, and we used the means and standard deviations of the wind field on each grid point across the ensemble members to model the uncertainty. Then we extracted uncertain LCSs through Monte Carlo particle tracing, finite-time Lyapunov exponents computation, and ridge detection as presented in [40]. The starting time of particle tracing was 00:00:00 UTC, August 27, 2015, and the advection time was 5 hours. Figures 16 (b) and (c) compare the visualization results of SDE and SPH with the spaghetti plots shown in Figure 16 (a) as reference. The radius of the sphere was set to 2 for SPH. To compare with SPH, the bandwidth σ was set to 0.67 for SDE calculation, such that the points on the surface with distances to the target location greater than 2 have negligible contribution to the density estimation result. In contrast with SPH, we use higher-order kernel functions to calculate surface densities, which gives more smooth results. Based on the visualization of SDE, regions near the Rocky Mountains that with upward and downward motions have higher likelihoods containing LCSs than other regions.

8.4 StreamSurfaces in Ensemble Flow Fields

We use SDE to explore streamsurfaces generated from a synthetic ensemble fluid flow dataset based on a 3D flow around a confined square cylinder [71]. The ensemble is obtained by injecting multiple realizations of noise in the vector field along the y direction. The noise samples are

generated by sampling 8 Gaussian kernels with shifted means and standard deviation 0.01, with 5 samples per kernel, to create an ensemble vector dataset with 40 members. An ensemble of streamsurfaces are then generated for a seeding curve close to the square cylinder. By applying the proposed bandwidth selection method, the bandwidth σ was set to 0.3. Figure 17 shows the visualization of the density estimation results generated based on the streamsurfaces. The streamsurfaces are visualized using spaghetti plots as shown in Figure 17 (a), and the SDE is visualized in Figure 17 (b). Compared with the streamsurfaces, we find that the SDE clearly reveals the variability of the streamsurfaces. Before the particles reach the square cylinder, the streamsurfaces are close to each other, shown in the SDE as high surface densities. After the particles reach the square cylinder, the flows are separated by the square cylinder, also highlighted in the SDE. The visualization of the SDE also highlights regions near the square cylinder on the right with relatively higher surface densities than regions far away from the square cylinder on the right, which can not be shown by overlaid rendering of the streamsurfaces due to occlusion. Figure 17 (c) visualizes two clusters of streamsurfaces as density fields selected in the hierarchical clustering tree, which highlight distinct flow features.

9 CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

In this work, we presented SDE to model positional uncertainty of surface features in 3D ensemble simulation data. SDE is computed based on surfaces represented as polygon meshes with no need of field datasets. Hence, our method can be applied on applications that no field datasets are available to model the positional uncertainty of the surfaces. We also presented a bandwidth selection method for SDE computation. Major trends and outlines of ensemble surfaces can also be extracted based on SDE. To this end, we transformed ensemble surfaces into density fields based on their contribution to SDE and organized the resulting density fields into a hierarchical representation based on their pairwise distances. We compared the proposed method with alternative density estimation methods in terms of accuracy and performance on synthetic ensemble surfaces that have known spatial distributions. We then demonstrated the effectiveness and usefulness of the proposed method for variate applications.

There are a few limitations of our approach that we would like to address in the future. First, we would like to reduce the computation time of SDE for interactive bandwidth selection and exploration. Second, the bandwidth selection method is limited to single smoothing for all directions currently. We would like to extend it to handle more complex bandwidth matrices. Third, the distance between density fields for clustering is limited to point-wise distance metrics for density values over the entire domain. In the future, we would like to study different distance metrics as well as considering sub-domains instead of the entire domain for clustering.

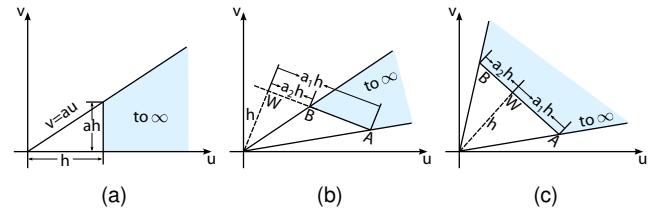


Fig. 18. (a) Region over which $\Gamma(h, a)$ gives the integral of the standard bivariate normal distribution. (b) and (c) Bivariate normal integral over areas defined by an edge of a given triangle.

APPENDIX

This appendix provides the method proposed by Owen [58] for calculating the bivariate normal integral over an area defined by an edge of a given triangle. He first defines a function $\Gamma(h, a)$ that gives the integral of the standard bivariate normal distribution over the area between line $v = au$ and $v = 0$ and to the right of line $u = h$, as shown in Figure 18 (a). $\Gamma(h, a)$ is then used to calculate the bivariate normal integral over an area defined by an edge of a given triangle, as the shaded areas shown in Figures 18 (b) and (c).

The function $\Gamma(h, a)$ is defined as

$$\Gamma(h, a) = \frac{\arctan(a)}{2\pi} - \frac{1}{2\pi} \sum_{i=0}^{\infty} c_i a^{2i+1}, \quad (17)$$

where

$$c_i = (-1)^i \frac{1}{2i+1} (1 - e^{-\frac{1}{2}h^2} \sum_{j=0}^i \frac{h^{2j}}{2^j j!}), \quad (18)$$

which converges rapidly for small a and h . To reduce runtime computations, we generate a lookup table for the Γ -function for h from 0 to 4.76, a from 0 to 1, and $a = \infty$, with all 0.01 in between. We combine the lookup table with bilinear interpolation to obtain the $\Gamma(h, a)$ for given a and h . For negative h or a , $\Gamma(-h, a) = \Gamma(h, a)$ and $\Gamma(h, -a) = -\Gamma(h, a)$. For $1 < a < \infty$, $\Gamma(h, a)$ is obtained by using

$$\Gamma(h, a) = \frac{1}{2}\Phi(h) + \frac{1}{2}\Phi(ah) - \Phi(h)\Phi(ah) - \Gamma(ah, \frac{1}{a}), \quad (19)$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution. For $h > 4.76$, $\Gamma(h, a) = 0$.

With the Γ -function, the integral of the standard bivariate normal distribution over an area defined by an edge of a given triangle can be obtained. For example, in Figure 18 (b), let AB be an edge of a triangle. The integral of the standard bivariate normal distribution over the shaded area, denoted as α_{AB} , can be obtained by using $\Gamma(h, a_1) - \Gamma(h, a_2)$ for $a_1 > a_2$, where h is the length between the origin and its projection W on the line defined by AB , which is

$$h = \frac{|u_1v_2 - u_2v_1|}{\sqrt{(u_2 - u_1)^2 + (v_2 - v_1)^2}}, \quad (20)$$

and the slopes a_1 and a_2 can be computed with

$$a_1 = \frac{|u_1(u_2 - u_1) + v_1(v_2 - v_1)|}{|u_1v_2 - u_2v_1|} \quad (21)$$

$$a_2 = \frac{|u_2(u_2 - u_1) + v_2(v_2 - v_1)|}{|u_1v_2 - u_2v_1|}, \quad (22)$$

where $(u_1, v_1)^T$ and $(u_2, v_2)^T$ are the coordinates of the vertices A and B . If W lies between A and B , then $\alpha_{AB} = \Gamma(h, a_1) + \Gamma(h, a_2)$, as shown in Figure 18 (c).

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

- [1] D. Thompson, J. A. Levine, J. C. Bennett, P. T. Bremer, A. Gyulassy, V. Pascucci, and P. P. Pébay, "Analysis of large-scale scalar data using hexels," in *Proceedings of 2011 IEEE Symposium on Large Data Analysis and Visualization*, 2011, pp. 23–30.
- [2] K. Pöthkow and H. C. Hege, "Positional uncertainty of isocontours: Condition analysis and probabilistic measures," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 10, pp. 1393–1406, 2011.
- [3] T. Pfaffelmoser, M. Reitinger, and R. Westermann, "Visualizing the positional and geometrical variability of isosurfaces in uncertain scalar fields," in *Proceedings of the 13th Eurographics*, 2011, pp. 951–960.
- [4] K. Pöthkow, B. Weber, and H.-C. Hege, "Probabilistic marching cubes," *Computer Graphics Forum*, vol. 30, no. 3, pp. 931–940, 2011.
- [5] K. Pöthkow, C. Petz, and H.-C. Hege, "Approximate level-crossing probabilities for interactive visualization of uncertain isocontours," *International Journal for Uncertainty Quantification*, vol. 3, no. 2, pp. 101–117, 2013.
- [6] T. Athawale and A. Entezari, "Uncertainty quantification in linear interpolation for isosurface extraction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2723–2732, 2013.
- [7] T. Athawale, E. Sakhaei, and A. Entezari, "Isosurface visualization of data with nonparametric models for uncertainty," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 777–786, 2016.
- [8] E. Sakhaei and A. Entezari, "A statistical direct volume rendering framework for visualization of uncertain data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 12, pp. 2509–2520, 2017.
- [9] A. Kumpf, B. Tost, M. Baumgart, M. Riener, R. Westermann, and M. Rautenhaus, "Visualizing confidence in cluster-based ensemble weather forecast analyses," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 109–119, 2018.
- [10] G. Haller, "A variational theory of hyperbolic Lagrangian coherent structures," *Physica D: Nonlinear Phenomena*, vol. 240, no. 7, pp. 547–598, 2011.
- [11] H. Obermaier and K. I. Joy, "Future challenges for ensemble visualization," *IEEE Computer Graphics and Applications*, vol. 34, no. 3, pp. 8–11, 2014.
- [12] J. Wang, S. Hazarika, C. Li, and H. W. Shen, "Visualization and visual analysis of ensemble data: A survey," *IEEE Transactions on Visualization and Computer Graphics*, 2018, Early Access.
- [13] K. Potter, A. Wilson, P.-T. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. R. Johnson, "Ensemble-vis: A framework for the statistical visualization of ensemble data," in *Proceedings of IEEE Workshop on Knowledge Discovery from Climate Data: Prediction, Extremes*, 2009, pp. 233–240.
- [14] M. N. Phadke, L. Pinto, O. Alabi, J. Harter, R. M. Taylor II, X. Wu, H. Petersen, S. A. Bass, and C. G. Healey, "Exploring ensemble visualization," Proc. SPIE, vol. 8294, no. 82940B, pp. 1–12, 2012.
- [15] J. Sanyal, S. Zhang, J. Dyer, A. Mercer, P. Amburn, and R. Moorhead, "Noodles: A tool for visualization of numerical weather model ensemble uncertainty," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1421–1430, 2010.
- [16] R. T. Whitaker, M. Mirzargar, and R. M. Kirby, "Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2713–2722, 2013.
- [17] M. Mirzargar, R. T. Whitaker, and R. M. Kirby, "Curve boxplot: Generalization of boxplot for ensembles of curves," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2654–2663, 2014.
- [18] M. Raj, M. Mirzargar, J. S. Preston, R. M. Kirby, and R. T. Whitaker, "Evaluating shape alignment via ensemble visualization," *IEEE Computer Graphics and Applications*, vol. 36, no. 3, pp. 60–71, 2016.
- [19] B. Zehner, N. Watanabe, and O. Kolditz, "Visualization of gridded scalar data with uncertainty in geosciences," *Computers & Geosciences*, vol. 36, no. 10, pp. 1268 – 1275, 2010.
- [20] F. Ferstl, K. Bürger, and R. Westermann, "Streamline variability plots for characterizing the uncertainty in vector field ensembles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 767–776, 2016.
- [21] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann, "Visual analysis of spatial variability and global correlations in ensembles of iso-contours," *Computer Graphics Forum*, vol. 35, no. 3, pp. 221–230, 2016.
- [22] ———, "Time-hierarchical clustering and visualization of weather forecast ensembles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 831–840, 2017.
- [23] I. Demir, J. Kehrer, and R. Westermann, "Screen-space silhouettes for visualizing ensembles of 3D isosurfaces," in *Proceedings of 2016 IEEE Pacific Visualization Symposium*, 2016, pp. 204–208.
- [24] A. Pang, C. Wittenbrink, and S. Lodha, "Approaches to uncertainty visualization," *The Visual Computer*, vol. 13, no. 8, pp. 370–390, 1997.
- [25] K. Brodlie, R. Allendes Osorio, and A. Lopes, "A review of uncertainty in data visualization," in *Expanding the Frontiers of Visual Analytics and Visualization*, J. Dill, R. Earnshaw, D. Kasik, J. Vince, and P. C. Wong, Eds. Springer London, 2012, pp. 81–109.
- [26] K. Potter, P. Rosen, and C. R. Johnson, "From quantification to visualization: A taxonomy of uncertainty visualization approaches," in *Uncertainty Quantification in Scientific Computing*. Springer, 2012, pp. 226–249.
- [27] P. J. Rhodes, R. S. Laramee, R. D. Bergeron, and T. M. Sparr, "Uncertainty Visualization Methods in Isosurface Rendering," in *Proceedings of Eurographics 2003 - Short Presentations*, 2003, pp. 83–88.
- [28] R. S. A. Osorio and K. W. Brodlie, "Contouring with uncertainty," in *Proceedings of Theory and Practice of Computer Graphics*, 2008, pp. 1–7.
- [29] S. Djurcicov, K. Kim, P. Lermusiaux, and A. Pang, "Visualizing scalar volumetric data with uncertainty," *Computers and Graphics*, vol. 26, no. 2, pp. 239–248, 2002.
- [30] G. Grigoryan and P. Rheingans, "Point-based probabilistic surfaces to show surface uncertainty," *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 5, pp. 564–573, 2004.
- [31] K. Pöthkow and H.-C. Hege, "Nonparametric models for uncertainty visualization," *Computer Graphics Forum*, vol. 32, no. 3, pp. 131–140, 2013.
- [32] S. Hazarika, A. Biswas, and H. W. Shen, "Uncertainty visualization using copula-based analysis in mixed distribution models," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 934–943, 2018.
- [33] C. M. Wittenbrink, A. T. Pang, and S. K. Lodha, "Glyphs for visualizing uncertainty in vector fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, no. 3, pp. 266–279, 1996.
- [34] M. Hlawatsch, P. Leube, W. Nowak, and D. Weiskopf, "Flow radar glyphs & static visualization of unsteady flow with uncertainty," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1949–1958, 2011.
- [35] S. K. Lodha, A. Pang, R. E. Sheehan, and C. M. Wittenbrink, "UFLOW: Visualizing uncertainty in fluid flow," in *Proceedings of IEEE Visualization 1996*, 1996, pp. 249–254.
- [36] R. P. Botchen, D. Weiskopf, and T. Ertl, "Texture-based visualization of uncertainty in flow fields," in *Proceedings of IEEE Visualization*, Oct 2005, pp. 647–654.
- [37] C. Petz, K. Pöthkow, and H.-C. Hege, "Probabilistic local features in uncertain vector fields with spatial correlation," *Computer Graphics Forum*, vol. 31, no. 3pt2, pp. 1045–1054, 2012.
- [38] H. Bhatia, S. Jadhav, P. T. Bremer, G. Chen, J. A. Levine, L. G. Nonato, and V. Pascucci, "Flow visualization with quantified spatial and temporal errors using edge maps," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 9, pp. 1383–1396, 2012.
- [39] G. Haller, "Lagrangian coherent structures from approximate velocity data," *Physics of Fluids*, vol. 14, no. 6, pp. 1851–1861, 2002.
- [40] H. Guo, W. He, T. Peterka, H. W. Shen, S. M. Collis, and J. J. Helmus, "Finite-time lyapunov exponents and Lagrangian coherent structures in uncertain unsteady flows," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 6, pp. 1672–1682, 2016.
- [41] T. Peterka, H. Croubois, N. Li, S. Rangel, and F. Cappello, "Self-adaptive density estimation of particle data," *SIAM Journal on Scientific Computing*, vol. 38, no. 5, pp. S646–S666, 2015.

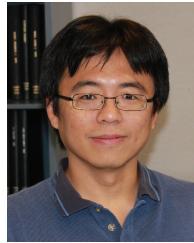
- [42] C. K. Birdsall and D. Fuss, "Cloud-in-cell computer experiments in two and three dimensions," University of California, Livermore. Lawrence Radiation Laboratory, Tech. Rep., 1969.
- [43] J. J. Monaghan, "Smoothed particle hydrodynamics," *Annual Review of Astronomy and Astrophysics*, vol. 30, no. 1, pp. 543–574, 1992.
- [44] W. E. Schaap and R. Van De Weygaert, "Continuous fields and discrete samples: reconstruction through delaunay tessellations," *Astronomy and Astrophysics*, vol. 363, pp. L29–L32, 2000.
- [45] W. E. Schaap, "Dtfe: the delaunay tessellation field estimator," Ph.D. dissertation, University of Groningen, The Netherlands, 2007.
- [46] C. K. Birdsall and D. Fuss, "Clouds-in-clouds, clouds-in-cells physics for many-body plasma simulation," *Journal of Computational Physics*, vol. 3, no. 4, pp. 494–511, 1969.
- [47] M. Zinsmaier, U. Brandes, O. Deussen, and H. Strobel, "Interactive level-of-detail rendering of large graphs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2486–2495, 2012.
- [48] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall, 1986.
- [49] C. Hurter, O. Ersoy, and A. Telea, "Graph bundling by kernel density estimation," *Computer Graphics Forum*, vol. 31, no. 3pt1, pp. 865–874, 2012.
- [50] J. Heinrich and D. Weiskopf, "Continuous parallel coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1531–1538, 2009.
- [51] H. Zhou, W. Cui, H. Qu, Y. Wu, X. Yuan, and W. Zhuo, "Splatting the lines in parallel coordinates," in *Proceedings of the 11th Eurographics*, 2009, pp. 759–766.
- [52] P. Muigg, M. Hadwiger, H. Doleisch, and M. E. Gröller, "Visual coherence for large-scale line-plot visualizations," *Computer Graphics Forum*, vol. 30, no. 3, pp. 643–652, 2011.
- [53] R. Scheepens, N. Willem, H. van de Wetering, and J. J. van Wijk, "Interactive visualization of multivariate trajectory data with density maps," in *Proceedings of 2011 IEEE Pacific Visualization Symposium*, 2011, pp. 147–154.
- [54] R. Scheepens, N. Willem, H. van de Wetering, G. Andrienko, N. Andrienko, and J. J. van Wijk, "Composite density maps for multivariate trajectories," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2518–2527, 2011.
- [55] O. D. Lampe and H. Hauser, "Interactive visualization of streaming data with kernel density estimation," in *Proceedings of 2011 IEEE Pacific Visualization Symposium*, 2011, pp. 171–178.
- [56] O. D. Lampe, J. Kehrer, and H. Hauser, "Visual analysis of multivariate movement data using interactive difference views," in *VMV 2010: Proceedings of Vision, Modeling, and Visualization*, 2010, pp. 315–322.
- [57] O. D. Lampe and H. Hauser, "Curve density estimates," *Computer Graphics Forum*, vol. 30, no. 3, pp. 633–642, 2011.
- [58] D. B. Owen, "Tables for computing bivariate normal probabilities," *The Annals of Mathematical Statistics*, vol. 27, no. 4, pp. 1075–1090, 1956.
- [59] T. G. Donnelly, "Algorithm 462: Bivariate normal distribution," *Commun. ACM*, vol. 16, no. 10, pp. 629–640, 1973.
- [60] M. Wand and M. Jones, *Kernel Smoothing*, ser. Monographs on statistics and applied probability. Chapman and Hall, 1995.
- [61] M. Rudemo, "Empirical choice of histograms and kernel density estimators," *Scandinavian Journal of Statistics*, vol. 9, no. 2, pp. 65–78, 1982.
- [62] A. W. Bowman, "An alternative method of cross-validation for the smoothing of density estimates," *Biometrika*, vol. 71, no. 2, pp. 353–360, 1984.
- [63] J. Kiefer, "Sequential minimax search for a maximum," *Proceedings of the American Mathematical Society*, vol. 4, no. 3, pp. 502–506, 1953.
- [64] W. He, X. Liu, H.-W. Shen, S. M. Collis, and J. J. Helmus, "Range likelihood tree: A compact and effective representation for visual exploration of uncertain data sets," in *Proceedings of 2017 IEEE Pacific Visualization Symposium*, 2017, pp. 151–160.
- [65] J. H. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.
- [66] C. Ericson, *Real-Time Collision Detection*. Boca Raton: CRC Press, Inc., 2004.
- [67] A. Knoll, Y. Hijazi, A. Kensler, M. Schott, C. Hansen, and H. Hagen, "Fast ray tracing of arbitrary implicit surfaces with interval and affine arithmetic," *Computer Graphics Forum*, vol. 28, no. 1, pp. 26–40, 2009.
- [68] P. F. J. Lermusiaux, "Uncertainty estimation and prediction for interdisciplinary ocean dynamics," *J. Comput. Phys.*, vol. 217, no. 1, pp. 176–199, 2006.
- [69] A. R. Robinson, "Realtime forecasting of the multidisciplinary coastal ocean with the littoral ocean observing and predicting system (loops)," in *Proceedings of Preprint Volume of the Third Conference on Coastal Atmospheric and Oceanic Prediction and Processes*, 1999, pp. 3–5.
- [70] C. Alexander, S. S. Weygandt, D. C. D. S. Benjamin, T. G. Smirnova, E. P. James, M. H. P. Hofmann, J. Olson, and J. M. Brown, "The high-resolution rapid refresh: Recent model and data assimilation development towards an operational implementation in 2014," in *Proceedings of 26th Conference on Weather Analysis and Forecasting / 22nd Conference on Numerical Weather Prediction*, 2014.
- [71] S. Camarri, M.-V. Salvetti, M. Buffoni, and A. Iollo, "Simulation of the three-dimensional flow around a square cylinder between parallel walls at moderate Reynolds numbers," in *XVII Congresso di Meccanica Teorica ed Applicata*, 2005, pp. 1000–1012.



Wenbin He is a Ph.D. student in computer science and engineering at the Ohio State University. He received his B.S. degree from the Department of Software Engineering at Beijing Institute of Technology in 2012. His research interests include visualization and analysis of large-scale scientific data, uncertainty visualization, and flow visualization.



Hanqi Guo is an assistant computer scientist in the Mathematics and Computer Science Division, Argonne National Laboratory. He received his Ph.D. degree in computer science from Peking University in 2014, and the B.S. degree in mathematics and applied mathematics from Beijing University of Posts and Telecommunications in 2009. His research interests are mainly in flow visualization, uncertainty visualization, and large-scale scientific data visualization.



interests are scientific visualization and computer graphics. He is a winner of the National Science Foundation's CAREER award and U.S. Department of Energy's Early Career Principal Investigator Award. He also won the Outstanding Teaching award twice in the Department of Computer Science and Engineering at the Ohio State University.



Tom Peterka is a computer scientist at Argonne National Laboratory, fellow at the Computation Institute of the University of Chicago, adjunct assistant professor at the University of Illinois at Chicago, and fellow at the Northwestern Argonne Institute for Science and Engineering. His research interests are in large-scale parallelism for in situ analysis of scientific data. His work has led to three best paper awards and publications in ACM SIGGRAPH, IEEE VR, IEEE TVCG, and ACM/IEEE SC, among others. Peterka received his Ph.D. in computer science from the University of Illinois at Chicago, and he currently works actively in several DOE- and NSF-funded projects.