BIVARIATE QUANTILE INTERPOLATION FOR ENSEMBLE DERIVED PROBABILITY DENSITY ESTIMATES

Brad Eric Hollister[*] & *Alex Pang*

Computer Science Department, Jack Baskin School of Engineering, 1156 High Street, University of California, Santa Cruz, California 95060, USA

Original Manuscript Submitted: 07/21/2014; Final Draft Received: 03/02/2015

Probability distribution functions (PDFs) may be estimated from members in an ensemble. For an ensemble of 2D vector fields, this results in a bivariate PDF at each location in the field. Vector field analysis and visualization, e.g., stream line calculation, require an interpolation to be defined over these 2D density estimates. Thus, a nonparametric PDF interpolation must advect features as opposed to cross-fading them, where arbitrary modalities in the distribution can be introduced. This is already achieved for 1D PDF interpolation via inverse cumulative distribution functions (CDFs). However, there is no closed-form extension to bivariate PDF. This paper presents one such direct extension of the 1D closed-form solution for bivariates. We show an example of physically coupled components (velocity) and correlated random variables. Our method does not require a complex implementation or expensive computation as does displacement interpolation Bonneel et al., ACM Trans. Graphics (TOG), 30(6):158, 2011. Additionally, our method does not suffer from ambiguous pair-wise linear interpolants, as does Gaussian Mixture Model Interpolation.

KEY WORDS: *spatial statistics, density estimation, computational statistics, random fields, uncertainty quantification, representation of uncertainty, spatial uncertainty*

1. INTRODUCTION

A fundamental operation used in most visualization algorithms is interpolation. Interpolation is used in workhorse visualization techniques such as marching cubes, direct volume rendering, and stream line generation, and many other popular algorithms. Performing interpolation is well defined when the data points and the interpolants are crisp. However, this is not the case when the data points consist of a distribution.

With increasing interest in representing uncertainty in modeling and simulation with techniques based on Monte Carlo methods, we are now faced with the challenge of analyzing and visualizing ensemble fields. Ensemble fields are made up of individual realizations, each a possible outcome, of the simulation. Assuming that the ensemble fields are defined over a regular Cartesian grid, a popular approach is to treat all the values at a given grid point from different realizations as a distribution. Recent works in this area have primarily assumed that the distribution follows a Gaussian distribution. Even more recent efforts have extended this to non-Gaussian distributions.

In this paper, we extend a closed-form 1D probability distribution function interpolation method [1] that advects features for nonparametric probability distribution functions (PDFs). It is essentially a method that interpolates quantiles of the corresponding cumulative distribution functions (CDFs) and then solves for the interpolant PDF. However, until now, there was no direct extension to bivariate distributions, which are needed to represent vector PDF interpolation.

This paper addresses both physical vector fields (e.g., velocity, angular momentum) and a vector of scalar fields (e.g., two scalar fields, for instance temperature and humidity in a vector representation). Our interpolation method

[*] Correspond to Brad Eric Hollister, E-mail: behollis@soe.ucsc.edu

2152–5080/15/$35.00

is general, and applies to both types of vectors. Our method is necessary for physical vector fields that cannot be decomposed into univariate distributions and for correlated random variables.

This work is motivated by the need for a nonparametric PDF interpolation that scales to large data sets by employing variable computational cost for required levels of accuracy. It is primarily applicable to multidimensional fields whose component random variables are correlated. Uncorrelated random variables may be treated as univariates.

2. RELATED WORK

A nice overview of statistical techniques for spatial interpolation was presented by Myers [2]. The techniques range from simple linear models with no covariance, to those using spatial structure functions. The survey however does not include nonparametric distribution interpolation. The paper does claim that interpolation is a solution to an inherently ill-posed problem, namely that it is a problem of prediction with limited data. For that, multiple models with different purposes can be employed. A more detailed survey [3], but focusing on geostatistical applications, compares methods according to different criteria such as local vs. global support, deterministic vs. stochastic, univariate vs multivariate, linear vs. nonlinear, etc. Among the methods that consider stochastic data, they assume normal distribution.

Within the visualization community, there are also a number of recent publications that address stochastic interpolation. Schlegel et al. [4] present a form of Kriging interpolation of spatial data for Gaussian distributions using a parameter-based approach. This technique relies on computing a covariance matrix and that the underlying data be formed from a Gaussian process. Pfaffelmoser et al. [5] visualize isosurfaces via a raycasting scheme, and perform spatial interpolation assuming the data have a Gaussian distribution at each location. Likewise, Pothkow et al. [6] discuss isocontour visualization of normally distributed data. They interpolated between grid points using the 0th and 1st moments without spatial correlation considerations. Their subsequent work [7] considered the effects of spatial correlation in visualizing isosurfaces using probabilistic marching cubes. An alternative method of looking at global correlation structures in a hierarchical fashion was presented in [8].

When data do not follow a Gaussian distribution, a more general uncertainty model is needed. Liu et al. [9] propose a Gaussian mixture to represent the distribution of voxel values in air temperature data. They perform volume rendering on the data set and interpolate between pairs of a fixed number of Gaussian components along cast rays. In their study, they found that four Gaussian kernels are sufficient for a variety of data sets that they examined. In addition, they support stationary and anisotropic correlations in the process, but at the expense of considering multimodal qualities of the probability distributions at grid points. For nonparametric representations of non-Gaussian distributions, operations on the distributions require different handling. Love, et al. [10] discuss two forms of a nonparametric interpolation method via convolution addition of probability distributions as well as bin-wise addition. Pohl, et al., [11] first transform the (discrete) distribution to Euclidean space via a set of Log Odds operations, where they can then be manipulated using conventional addition and multiplication. Results are then mapped back to probabilistic space via a reversible transform.

Uncertainty in vector fields is of great interest to at least two broad fields: environmental science, e.g., oceanography and meteorology [12, 13], and fiber tracking of diffusion tensor magnetic resonance images (DT MRI). Both [14] and [15], discuss non-Gaussian methods in these areas of research. Otto et al. present analysis of 2D [16] and 3D velocity fields [17] using particle advection, critical points, and segmentation of field topology. Petz et al. [18] also analyze uncertain velocity fields modeled as Gaussian random fields with spatial correlation.

There is a growing body of work on probabilistic fiber tracking. Unlike velocity fields, the tracks here represent fiber connectivity from one region to another and are obtained by integrating the major eigenvector field of symmetric DT MRI data set. The main source of uncertainty can be attributed to inadequate resolution in the data acquisition stage. However, there are numerous other sources as well [19]. While most of the earlier works on probabilistic fiber tracking delved on the inadequacy of the simple tensor representation to show alternative trajectories due to multiple fiber populations within a cell, more recent works are based on high angular resolution diffusion imaging (HARDI) data, which makes it is possible to describe fiber orientations using more sophisticated formulations such as spherical harmonics and multitensor representations. In a recent paper, Jiao et al. [20] describe a local, icon-based presentation of an ensemble field of fiber orientation distribution functions (ODF). The results of our paper can be used toward spatial analysis of such ensemble fields, for example.

There is much interest in the meteorological community to provide better visualization of forecast data. Slingsby et. al [21], discuss how users interpret and use weather data, specifically hurricane data. Storm path information are examined from historical data. They draw attention to spatial and temporal clustering and its undervalued status among those currently employing such visualization software. Weather forecasts are usually based on an ensemble of predictions. For that, Potter et al. [22] describe a framework for viewing stochastic information from ensembles. This package allows for visualization of spaghetti plotting, etc. of weather data. Sanyal et al. [23] present Noodles, a software package for displaying uncertainty in stream lines and other weather data visualization for ensemble forecasting. Potter et al. [24] describe a software tool to visualize two-dimensional sets of distribution data. It displays a contour of field PDF values and allows for a normed difference between data PDFs and an ansatz selected by the user. More recently, Phadke et al. [25] present two novel visualization methods for ensembles. Primarily, they allow simultaneous viewing of multiple ensemble members. They also present a technique called *Screen Door Tinting* which applies value changes to field points that show differences between ensembles.

From the point of view of users, Martin et al. [26] point out the difficulty of users to identify hurricane directional movement and speed from current data visualization, or directly on vector fields. In a similar study, Broad et al. [27], further emphasize interpretation and usage of complex weather data. They show how a general interpretation of a Gaussian distribution of hurricane direction prediction can lead to inaccurate views on the probability within a *cone of uncertainty*. Clearly, if multimodal velocity distribution is calculated with such a broad region of uncertainty using a Gaussian assumption, incorrect estimation of the probability of hurricane direction can occur, most specifically within the general population who can be greatly impacted by such interpretation. A non-Gaussian consideration for vector field visualization together with a redesigned visualization may rectify this issue to a degree.

The method presented by Liu et al. [9], which proposes a Gaussian mixture model, is insufficient for bivariate PDF. Despite the use of a fixed number of Gaussian basis functions for PDF estimates, the interpolation is only unambiguous for 1D PDF when pairing Gaussian components by the order of their mean parameter. For 2D Gaussian mixture models, there is no such ordering. It is possible to order bivariate Gaussian components based on their mean probability, but this does not follow from the 1D case of ordering based on the mean parameter value.

Displacement Interpolation, developed by Bonneel et al. [28], is a general method for multivariate PDF interpolation. It is shown to reduce to the 1D PDF interpolation presented by Read [1]. It satisfies the advection of features by interpolating populations instead of cross-fading them. (Bonneel et al. provide an in-depth discussion of this property in their paper.) It is based on solving for intermediate solutions to the Earth Mover's Distance, a minimum cost problem of transforming one PDF into another. This method does not scale well to 2D field interpolation, however. It is computationally costly, with current CPU implementations (using compiled code) taking on the order of minutes to hours for interpolation between only two PDF. In the form presented by Bonneel et al., it is developed only for interpolation between two PDF.

3. BIVARIATE QUANTILE INTERPOLATION

3.1 Derivation

We extend a CDF-based interpolation method for use with bivariate PDF, which is needed for uncertain 2D velocity fields. The original 1D method was analytically derived in [1], and is shown below. Here, $F(x)$ is the CDF with its associated PDF, $f(x)$, as in Eq. (1).

$$F(x) = \int_{-\infty}^{x} f(h)dh \tag{1}$$

f_0 and f_1 are two known PDF used for the interpolation. Their CDF are F_0 and F_1, respectively. The quantile y corresponds to both x_0 and x_1 in Eqs. (2) and (3).

$$F_0(x_0) = y \tag{2}$$

$$F_1(x_1) = y \tag{3}$$

$\bar{F}(\bar{x})$ is the interpolant CDF found from linearly interpolating between x_0 and x_1, shown in Eqs. (4) and (5).

$$\bar{x} = (1-\alpha)x_0 + \alpha x_1 \tag{4}$$

$$\bar{F}(\bar{x}) = y \tag{5}$$

Using F^{-1}, we have $F_0^{-1}(y) = x_0$, $F_1^{-1}(y) = x_1$ and $\bar{F}^{-1}(y) = \bar{x}$. Substituting these results into Eq. (4) yields

$$\bar{F}^{-1}(y) = (1-\alpha)F_0^{-1}(y) + \alpha F_1^{-1}(y) \tag{6}$$

Knowing that $dx = dF^{-1}(y)$, $dy = dF(x)$ and $dx/dy = (dy/dx)^{-1}$, we have

$$\frac{dF^{-1}(y)}{dy} = \left[\frac{dF(x)}{dx}\right]^{-1} = \frac{1}{f(x)} \tag{7}$$

Thus, applying d/dy to Eq. (6), and solving for $\bar{f}(\bar{x})$ produces

$$\bar{f}(\bar{x}) = \frac{f_0(x_0)f_1(x_1)}{(1-\alpha)f_1(x_1) + \alpha f_0(x_0)} \tag{8}$$

Our contribution is the novel extension to 2D PDF interpolation. Equation (9) represents the 2D conceptual extension of Eq. (8). The parameter $t \in [0,n]$ is introducted to provide a unique one-to-one correspondence between x and y pairs on the corresponding quantile curves from two bivariate PDFs f_0 and f_1, the known PDFs we interpolate from.

$$\bar{f}(\bar{x}(t_i), \bar{y}(t_i)) = \frac{f_0(x_0(t_i), y_0(t_i))f_1(x_1(t_i), y_1(t_i))}{(1-\alpha)f_1(x_1(t_i), y_1(t_i)) + \alpha f_0(x_0(t_i), y_0(t_i))} \tag{9}$$

Additionally, α is the linear interpolation factor that determines the Euclidean distance in the scaled probability space of the interpolant $(\bar{x}(t_i), \bar{y}(t_i))^T$. This relationship is expressed in Eq. (10).

$$\begin{bmatrix} x_0(t_i) \\ y_0(t_i) \end{bmatrix} + \alpha \begin{bmatrix} x_1(t_i) - x_0(t_i) \\ y_1(t_i) - y_0(t_i) \end{bmatrix} = \begin{bmatrix} \bar{x}(t_i) \\ \bar{y}(t_i) \end{bmatrix} \tag{10}$$

The parameter t, is taken as the fraction of the arc length of the rectified quantile curves from f_0 and f_1. The arc length L of curve C is defined as in Eq. (11) on the interval $[a,b]$. $ds^2 = dx^2 + dy^2$ for the infinitesimal line segment ds.

$$L(C) = \int_a^b ds = \int_a^b \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx \tag{11}$$

For finite numerical approximations, where C is the image of a continuous function $l : [a,b] \to \mathbb{R}^n$, we have

$$L(C) = \sup_{a=t_0<t_1<\cdots<t_n=b} \sum_{i=0}^{n-1} d(l(t_i), l(t_{i+1})) \tag{12}$$

All quantile curves are indexed with the same number n of finite t_i, regardless of the value of $L(C)$. Effectively then, each $(x(t_i), y(t_i))^T$ pair between curves are the same fractional length of their curve.

Our method does not seek to minimize various metrics placed on mapped curve segments. For instance, we do not minimize distance in the sample space between paired samples on the quantile curves being interpolated but use the simpler heuristic of arc length parameterization.

For interpolation within a grid cell, Eq. (9) can be extended using bilinear interpolation via both α and β weights for the orthogonal directions of the grid. The α and β weights within the unit cell are shown in Fig. 1.

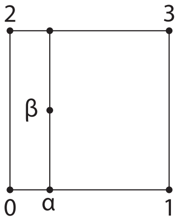

FIG. 1: Unit cell interpolation using both α and β.

In Eq. (13), we show the interpolation solved for the unit cell case. For brevity, we omit the $(x(t_i), y(t_i))$ pairs associated with each PDF. Each vertex represents the estimated PDFs from the ensemble for those locations. Setting either α or β to zero reduces to interpolation along a line.

$$\bar{f} = \frac{f_0 f_1 f_2 f_3}{f_1 f_2 f_3 + \alpha A + \beta B + \alpha\beta C} \tag{13}$$

A, B, and C are shown in Eqs. (14)–(16), respectively.

$$A = f_0 f_2 f_3 - f_1 f_2 f_3 \tag{14}$$

$$B = f_0 f_1 f_3 - f_1 f_2 f_3 \tag{15}$$

$$C = f_1 f_2 f_3 - f_0 f_2 f_3 - f_0 f_1 f_3 - f_0 f_1 f_2 \tag{16}$$

3.2 Algorithm

The major steps of the quantile interpolation method are shown in Fig. 2.

Stages *gather samples* and *estimate density* are implementation specific. We do not cover their implementation details here and the user may choose varying approaches depending on the data. For example, kernel density estimation (KDE) [29] with different window settings can be used for density estimation.

For the *CDF calculation* stage, we collect (u, v) pairs for each requested quantile curve. We use u and v to refer to the components of a 2D velocity vector, in place of x and y from the previous section. The input to the routine is a data object that represents the density estimate. The object supports returning the maximum and minimum values for u and v and the density for given extents. The routine is shown in Algorithm 1.

$quantiles$ is the set of quantiles. $dobj$ is the density object. $qpts$ is a dictionary of point lists, whose key is a quantile from $quantiles$ and whose value is a list of points. Each point is a (u, v) pair on the corresponding quantile

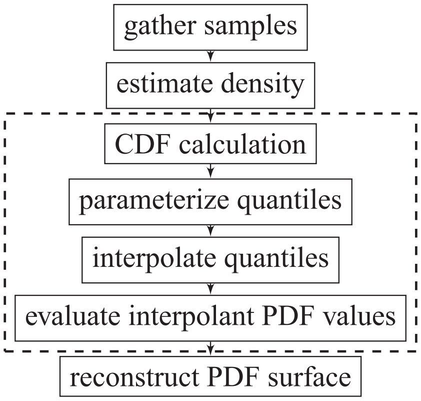

FIG. 2: Quantile PDF interpolation method. Dashed outline signifies core method stages discussed in the paper.

Algorithm 1: CDF calculation

```
Data: $dobj$, $quantiles$
Result: $qpts$
initialize $qpts$;
for $i = dobj.min\_u$ to $dobj.max\_u$ do
    for $j = dobj.min\_v$ to $dobj.max\_v$ do
        $d =$ density in region $(min\_u, min\_v)$ to $(i, j)$;
        foreach $q$ in $quantiles$ do
            if $q - \mathrm{TOL} \leq d < q + \mathrm{TOL}$ then
                $qpts[q].append((i, j))$;
            end
        end
    end
end
```

within a tolerance TOL. TOL=$1 \div 2|quantiles|$, where $|quantiles|$ is the cardinality of $quantiles$. The intervals from $dobj.min\_u$ to $dobj.max\_u$, and from $dobj.min\_v$ to $dobj.max\_v$, are both divided evenly by DIV, the number of divisions along each dimension. DIV can be tuned for desired resolution and CPU timings. We show our choice for DIV in Table 1 under *integration mesh size*.

For the *parameterize quantiles* stage (see Algorithm 2), we iterate through each member of $qpts$ and interpolate each individual curve using a curve object $cobj$, that can later be evaluated to obtain any v indexed by u.

This routine returns $qcurves$, a list of points from a parameterization of a curve represented by $cobj$. We approximate the parameter t_i in Eq. (12) by evenly dividing the entire interval of a quantile curve from an ortho-projection onto the u axis by NUM_PTS (the number of points chosen for parameterization). We then evaluate the $cobj$ from this interval of u values. We assume that the quantile curves are monotonically increasing over the interval.

For *interpolate quantiles* see Algorithm 3. This routine loops through all members of $quantiles$ and interpolates each parameterized point between corresponding quantiles. $iqcurves$ store the interpolated points for interpolant

TABLE 1: Parameters used bivariate quantile interpolation

Integration mesh size	200×200
Number of quantiles	$\leq$100
Quantile curve interpolation	Linear
Number of points per quantile	150
PDF surface interpolation	Linear simplicial

Algorithm 2: Parameterize quantiles

```
Data: $qpts$, $quantiles$
Result: $qcurves$
initialize $qcurves$;
foreach $q$ in $quantiles$ do
    $cobj =$ interpolate curve for all $pts$ in $qpts[q]$;
    foreach $u$ in evenly spaced NUM_PTS over interval $[qpts[q][0], qpts[q][index\ at\ list\ length - 1]]$ do
        $qcurves[q].append((u, cobj(u))$;
    end
end
```

Algorithm 3: Bilinear interpolation of quantile curves

```
Data: $dobj0$, $dobj1$, $dobj2$, $dobj3$, $qcurves0$, $qcurves1$, $qcurves2$, $qcurves3$, $\alpha$, $\beta$, $quantiles$
Result: $ipdf$
initialize $iqcurves01$;
initialize $iqcurves23$;
initialize $ipdf$;
foreach $q$ in $quantiles$ do
    foreach $idx$ in $qcurves0[q]$ do
        $vec01 = qcurves1[q][idx] - qcurves0[q][idx]$;
        $iqcurves01[q].append(\alpha * vec01)$;
    end
    foreach $idx$ in $qcurves2[q]$ do
        $vec23 = qcurves3[q][idx] - qcurves2[q][idx]$;
        $iqcurves23[q].append(\alpha * vec23)$;
    end
    foreach $idx$ in $qcurves01[q]$ do
        $vec = qcurves23[q][idx] - qcurves01[q][idx]$;
        $ipt = \beta * vec$;
        $idens$ = evalPDF($dobj0$,$dobj1$,$dobj2$,$dobj3$,$ipt$,$\alpha$,$\beta$);
        $ipdf.append((ipt.u, ipt.v, idens))$;
    end
end
```

quantiles. Using Eq. (10), we calculate $vec01$ and $vec23$. vec follows in a similar fashion for β. $dobj0$, $dobj1$, $dobj2$, and $dobj3$ are the density objects associated with each unit cell vertex in Fig. 1. $ipdf$ is returned and is a list of surface points on the interpolated PDF.

The *evalutate PDF values* stage is a direct calculation using Eq. (13), invoked during *interpolate quantiles* as the method *evalPDF*.

For the final *reconstruct PDF* step, a reconstruction of the PDF surface is performed using a suitable interpolation such as those available in SciPy [30] for irregular grid data. In this study, we tessellate the input point set to three-dimensional simplices, and interpolate linearly on each simplex.

4. RESULTS

Our implementation was written in Python, utilizing the SciPy package. All the computations were performed on the CPU. The computer system used for running the experiments was an Intel Core i7-3930k with 32 GB of RAM.

4.1 Synthetic Data

We construct a toy example consisting of a unimodal and bimodal distribution. Our mean parameter(s) for the 2D PDFs are the mean vector $\mu_i = (u, v)^T$, where u and v are the components aligned with the Cartesian x-y coordinate system. Spherical covariance matrices are used, i.e., the covariance matrix designation is a multiple of the identity matrix. The number of samples drawn from each distribution is 600 when estimating the PDF for interpolation.

The unimodal distribution is defined as

$$\mathcal{N}_1(\mu_1, \Sigma_1), \mu_1 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{17}$$

The bimodal distribution is the sum of two bivariate normals, where the first is weighted 0.6 and the second is weighted 0.4:

$$\mathcal{N}_3(\mu_3, \Sigma_3), \mu_3 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \Sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{18}$$

$$\mathcal{N}_4(\mu_4, \Sigma_4), \mu_4 = \begin{bmatrix} -2 \\ -1 \end{bmatrix}, \Sigma_4 = \begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix} \tag{19}$$

The parameters used for interpolation are in Table 1. See Table 2 for CPU timings. The results of interpolating between the synthetic PDF are shown in Fig. 3.

4.2 Application

Our ensemble data set covers a region of the Massachusetts Bay on the east coast of the United States of America [10] and is provided by Dr. Lermusiaux from MIT. The Massachusetts Bay volume in the study was divided into 53 × 90 grid with 16 depths. The depths at these 53 × 90 grid points vary significantly: depths as shallow as 90 meters and as deep as 196 meters were recorded. Our data are representative of environmental studies discussed in [12, 13].

We apply bivariate quantile interpolation to selected grid points over the spatial domain. We sub-sample at a quarter of the resolution of the original data, and keep the hidden data points as the "known" distribution to compare against our interpolants at $\alpha = 0.5$. For velocity fields, it is possible to interpolate over the temporal-domain as well. For instance, one could choose the same grid point but two different time steps. Additionally, it is possible to interpolate over space and time. The interpolation is general and applicable to multiple scenarios. However, in this study, we show interpolation between velocity PDF separated by space for the same value of time.

We choose two pairs of representative examples from the data for velocity. The first pair is an interpolation well within the boundaries of the data set (at a depth of 90 meters). The second pair is an interpolation that includes multimodal distributions but is along the boundary of the data set (at the same depth level). These interpolations are shown in Figs. 4 and 5 and referred to pair 1 and pair 2 in Tables 3 and 4.

A third pair of PDF use a vector of temperature and salt concentration (see Fig. 6 and Tables 3, 4). The interpolation was performed at the same spatial location as the first pair of PDF. These variables were tested for correlation using the Spearman rank-order correlation coefficient and the p-value to test for noncorrelation [31]. For our data these are $\rho = -0.3093$ and p-value $= 8.946 \times 10^{-15}$.

Our metric for the variation between an interpolant and the known grid point density estimate is Earth Mover's Distance (EMD). EMD is a linear optimization initially developed for supply-demand transportation. EMD minimizes the cost of transforming one PDF into another by moving mass from one PDF to the other [32]. The transformation cost between two PDF P and Q is expressed by the following formulation:

$$EMD(P,Q) = min_{\{F=f_{ij}\}} \frac{\Sigma_{i,j} f_{ij} d_{ij}}{\Sigma_{i,j} f_{ij}} \tag{20}$$

where d_{ij} is a pre-defined ground distance between supplier i and consumer j, and $F = f_{ij}$ is a set of flows which defines the amount of mass transported from supplier i to consumer j. We use OpenCV's implementation of EMD, with the L2-distance parameter [33].

The EMD measured for our interpolation examples are listed in Table 3. CPU performance is listed in Table 4.

TABLE 2: Average CPU timings (in seconds) in toy example

CDF calculation	139.92
Quantile curve parameterization	0.02
Quantile curve interpolation	1.17
Interpolant PDF evaluation	6.22
PDF surface reconstruction	1.18

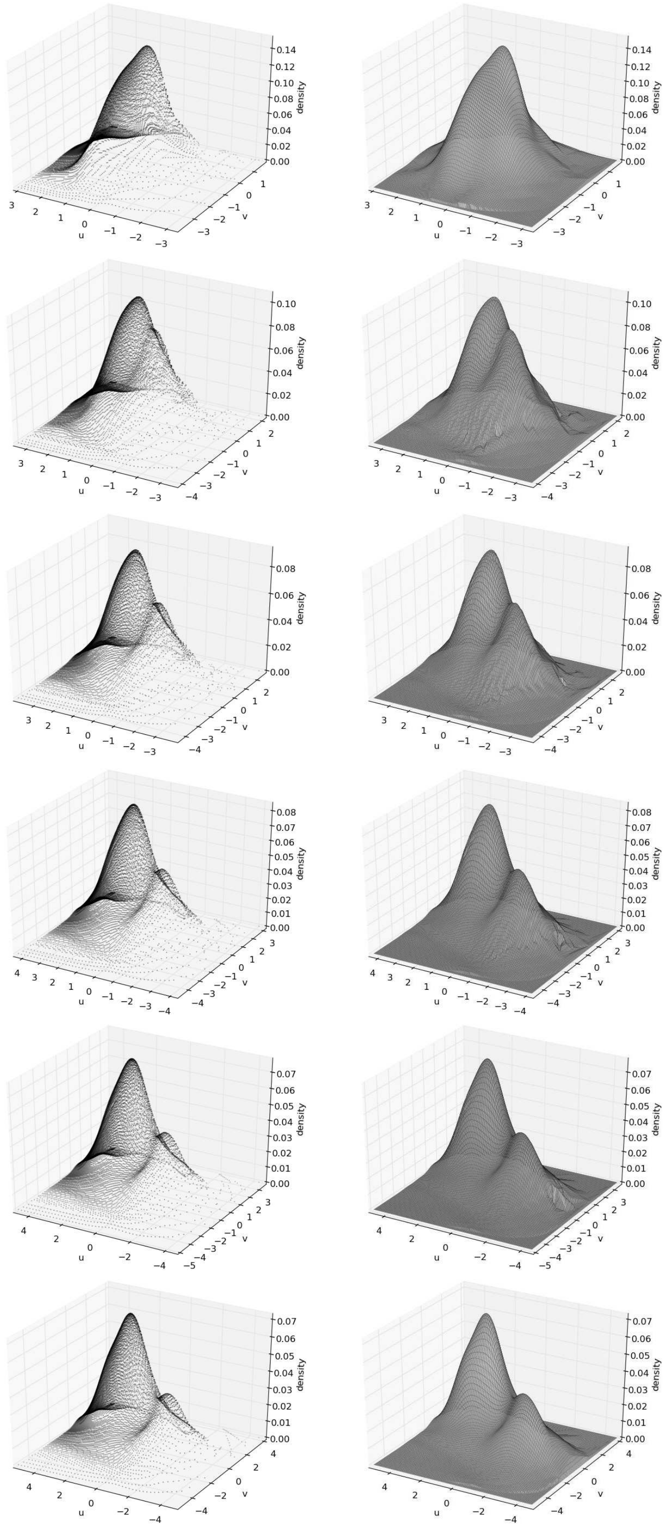

FIG. 3: Interpolation from top ($\alpha = 0.0$) to bottom ($\alpha = 1.0$). Left column without surface interpolation. Right column with surface interpolation.

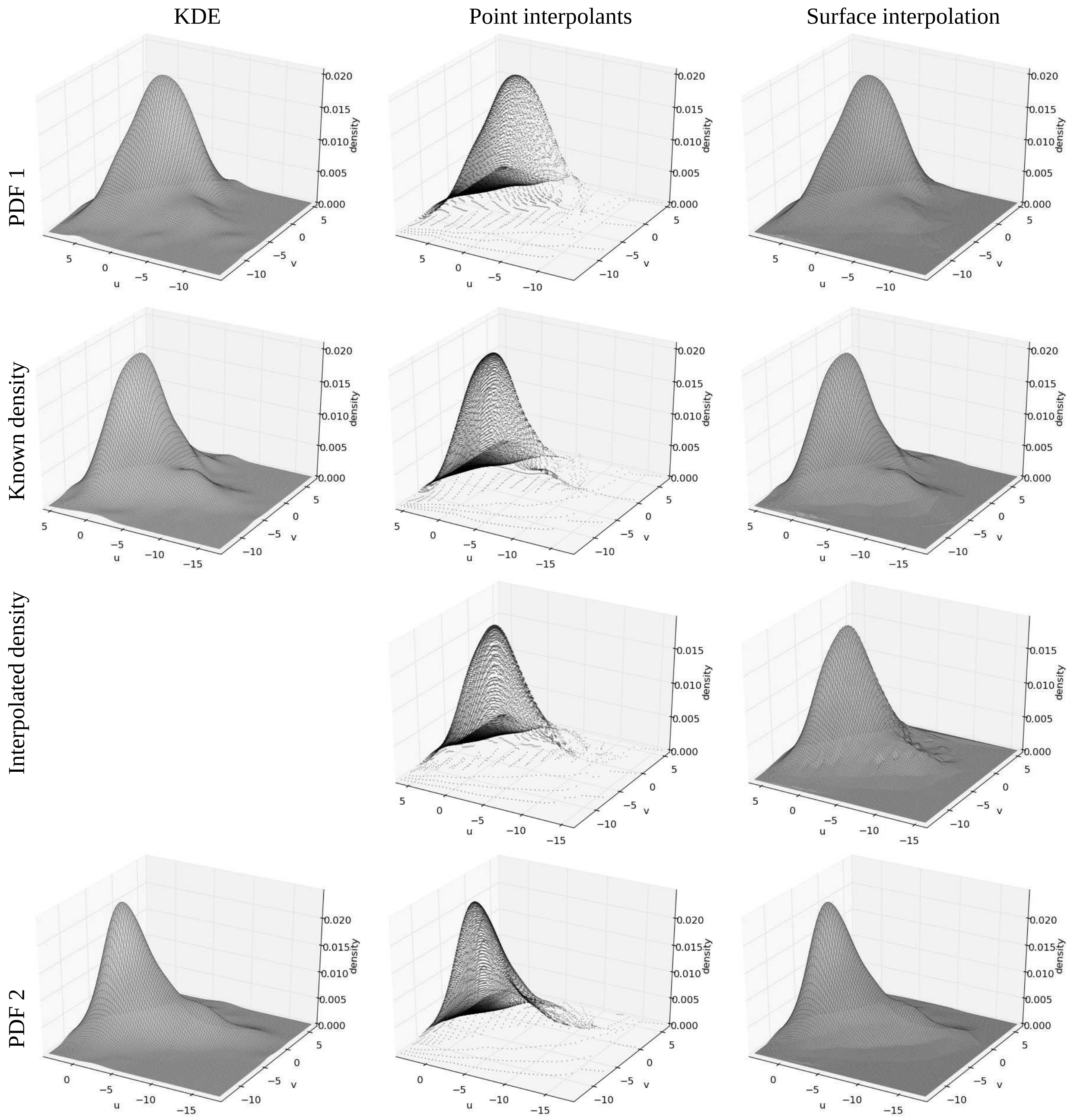

FIG. 4: Pair 1 for simulation data using velocity components. Green distributions represent KDEs at grid points in data set. Blue distributions represent results of interpolation. The top row (PDF 1) and bottom row (PDF 2) contain the known distributions used for interpolation. We compare the second row density estimate with the third row containing the interpolant density.

5. DISCUSSION

EMD values measured in table 3 show good results for pair 1, but the EMD measurement is higher for pair 2. The densities in pair 1 follow a smoother transition, while we miss the bimodal distribution in the second example. Minimal

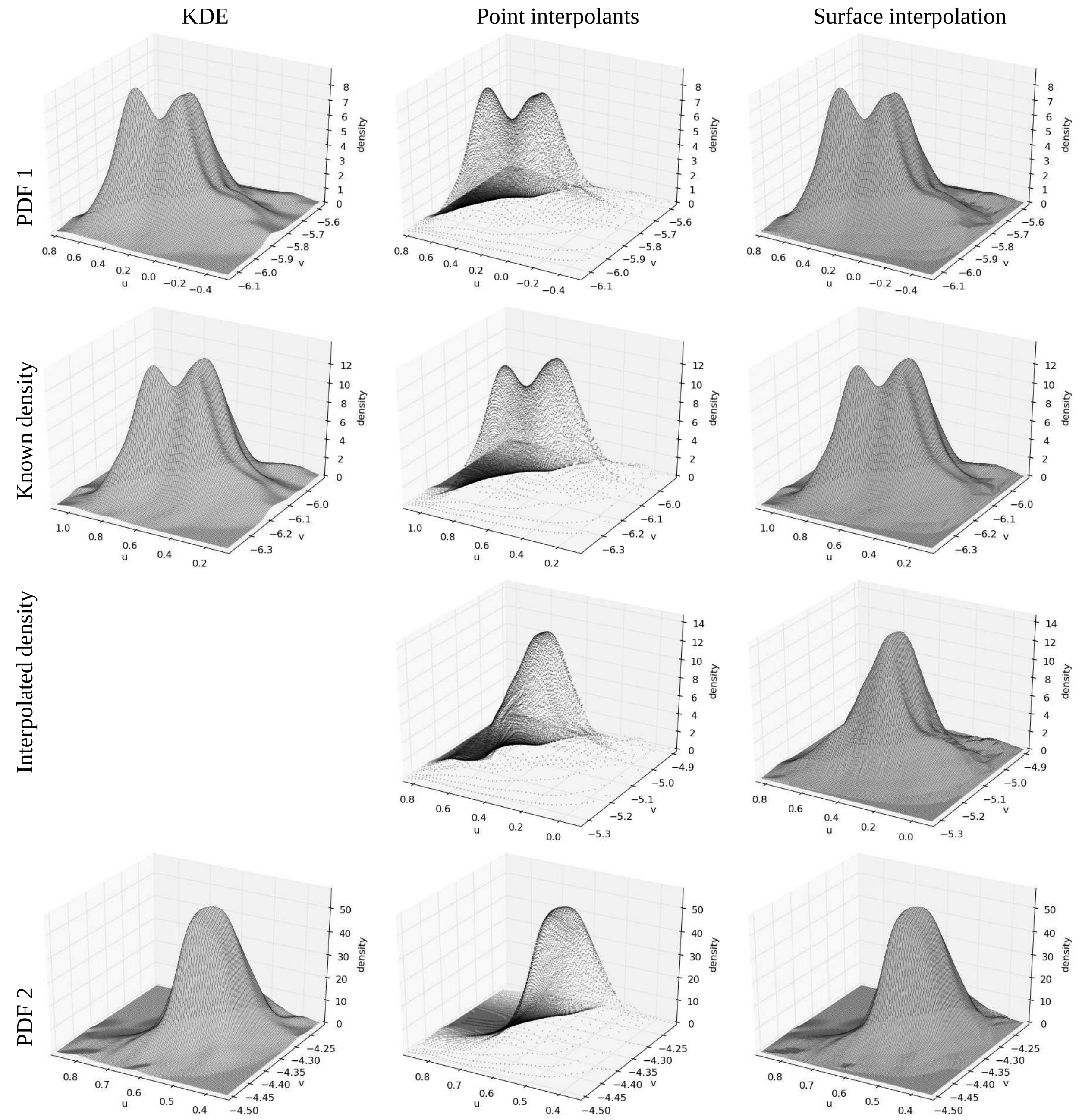

FIG. 5: Pair 2 for simulation data using velocity components. Green distributions represent KDEs at grid points in data set. Blue distributions represent results of interpolation. The top row (PDF 1) and bottom row (PDF 2) contain the known distributions used for interpolation. We compare the second row density estimate with the third row containing the interpolant density.

EMD difference is measured for surface interpolation alone in both cases. Pair 3 has slightly higher EMD values overall, while there is still good agreement with the known and interpolated distributions. This discrepancy is likely due to the slight clipping of the KDE (range values) versus the fill value of zero for surface interpolation.

TABLE 3: Earth mover's distance measurements for simulation data shown in Figs. 4–6. We compute EMD for the interpolant at $\alpha = 0.0$ in the entries of the row labeled PDF 1. Similarly, we compute EMD at $\alpha = 1.0$ in the row labeled PDF 2

Distribution	Pair 1	Pair 2	Pair 3
PDF 1	0.397509	0.473298	1.040877
Know PDF	0.317712	0.290578	0.351446
Interpolated PDF	1.362007	3.958944	5.588929
PDF 2	0.542557	0.729264	0.568048

TABLE 4: CPU timings (in seconds) for simulation data

Algorithm Stage	Pair 1	Pair 2	Pair 3
Avg. CDF calculation per PDF	154.34	135.88	84.403
Parameterization	0.010	0.010	0.017
Curve interpolation	1.29	1.28	0.679
Interpolant PDF evaluation	6.17	6.09	5.678
PDF surface reconstruction	1.14	1.31	0.979

It was also found that the number of samples in a density estimate increase CPU time for CDF calculation. This is due to the underlying implementation of SciPy and is not addressed in this paper.

Various increases in efficiency could be gained by porting the density object implementation to compiled code. Interpolating points from quantile curves in parallel on the GPU is another possible way to decrease execution time. An implementation may also be chosen to store quantile curve calculations for increased interpolation efficiency.

Note that the number of quantiles may be less than or equal to the number requested, as shown in Table 1. A chosen CDF integration mesh resolution is not always sufficient to capture the requested number of quantiles for a given distribution. Our implementation uses a fixed CDF integration mesh resolution.

Relaxing the assumption that quantile curves monotonically increase might allow better interpolation for cases where this is not always true. However, most distributions that we study have densities where this assumption is valid. In any case, this generality in the algorithm would increase execution time.

The interpolation methods presented in this paper do not account for spatial covariance with surrounding grid point distributions. We interpolate unique surface values of individual PDFs which do not relate as a whole to surrounding PDFs when considered in isolation.

Interpolation is inherently ill-posed. The quality of the interpolants are dependent on the smoothness of the underlying field. Therefore, procedures for measuring the smoothness of ensemble data sets are important here, but also for calculating probabilistic gradient fields. Such gradients are not easily defined for ensembles using finite difference. In any case, such analysis constitutes further study.

6. CONCLUSION

We presented a direct extension of 1D PDF interpolation using quantile interpolation for the bivariate case. This interpolation is useful for interpolating within random fields whose components are inseparable, such as 2D velocity PDF and other correlated random variables. Further studies of visualization using interpolation will be facilitated by such interpolation. Under circumstances where multiple scalar fields are interpolated, a univariate approach is best for performance and to reduce over-smoothing in density estimation [34] if the fields are uncorrelated.

While Gaussian mixture model interpolation is ambiguous with respect to its pair-wise interpolants, we have provided a better alternative. For 2D vector fields, bivariate quantile interpolation is faster than displacement interpolation and can be more easily implemented.

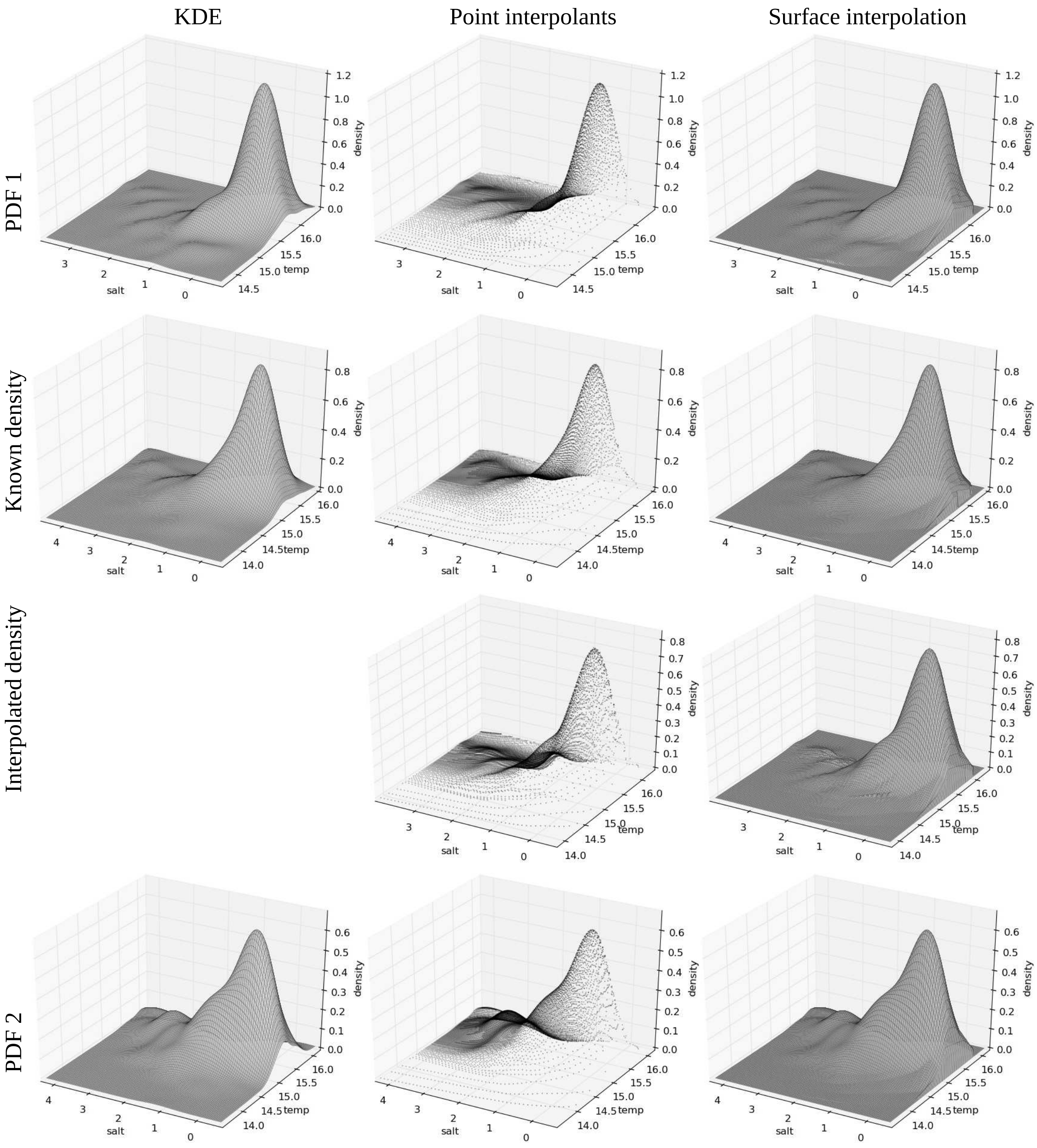

FIG. 6: Pair 3 for simulation data using temperature and salt concentration. Green distributions represent KDEs at grid points in data set. Blue distributions represent results of interpolation. The top row (PDF 1) and bottom row (PDF 2) contain the known distributions used for interpolation. We compare the second row density estimate with the third row containing the interpolant density.

ACKNOWLEDGMENTS

The authors wish to thank Dr. Pierre Lermusiaux and his group at MIT for sharing the ensemble data set. The authors also wish to thank Mitchell Allen for reproducing and independently testing our method.

REFERENCES

1. Read, A., Linear interpolation of histograms, *Nucl. Instrum. Methods Phys. Res.*, 425:357–360, 1999.
2. Myers, D., Spatial interpolation: An overview, *Geoderma*, 62(1-3):17–28, 1994.
3. Li, J. and Heap, A., A review of spatial interpolation methods for environmental scientists, Tech. Rep. GeoCat 68229, Geoscience Australia, 2008.
4. Schlegel, S., Korn, N., and Scheuermann, G., On the interpolation of data with normally distributed uncertainty for visualization, *IEEE Trans. Visualiz. Comput. Graphics*, 18(12):2305–2314, 2012.
5. Pfaffelmoser, T., Reitinger, M., and Westermann, R., Visualizing the positional and geometrical variability of isosurfaces in uncertain scalar fields, *Euro. IEEE Symp. Visual.*, 30(3):951–960, 2011.
6. Pöthkow, K. and Hege, H.-C., Positional uncertainty of isocontours: Condition analysis and probabilistic measures, *IEEE Trans. Visualiz. Comput. Graphics*, 17(10):1393–1406, 2011.
7. Pöthkow, K., Weber, B., and Hege, H.-C., Probabilistic marching cubes, In *Proceedings of the 13th Eurographics/IEEE—VGTC Conference on Visualization*, EuroVis'11, Aire-la-Ville, Switzerland, pp. 931–940, 2011.
8. Pfaffelmoser, T. and Westermann, R., Visualization of global correlation structures in uncertain 2D scalar fields, *Comput. Graphics Forum*, 31:1025–1034, 2012.
9. Liu, S., Levine, J. A., Bremer, P.-T., and Pascucci, V., Gaussian mixture model based volume visualization, *IEEE Symposium on Large Data Analysis and Visualization*, pp. 73–77, 2012.
10. Love, A., Kao, D. L., and Pang, A., Visualizing spatial multivalue data, *IEEE Comput. Graphics Appl.*, 25(3):69–79, 2005.
11. Pohl, K. M., Fisher, J., Bouix, S., Shenton, M., McCarley, R. W., Grimson, W. E., Kikinis, R., and Wells, W. M., Using the logarithm of odds to define a vector space on probabilistic atlases, *Med. Image Anal.*, 11:465–477, 2007.
12. Lermusiaux, P. F., Chiu, C.-S., Gawarkiewicz, G. G., Abbot, P., Robinson, A. R., Miller, R. N., Haley, P. J., Leslie, W. G., Majumdar, S. J., Pang, A., and Lekien, F., Quantifying uncertainties in ocean predictions, Tech. Rep., DTIC Document, 2006.
13. Lermusiaux, P. F., Uncertainty estimation and prediction for interdisciplinary ocean dynamics, *J. Comput. Phys.*, 217(1):176–199, 2006.
14. Schoelzel, C. and Friederichs, P., Multivariate non-normally distributed random variables in climate research–introduction to the copula approach, *Nonlin. Processes Geophys.*, 15(5):761–772, 2008.
15. Bocquet, M., Pires, C. A., and Wu, L., Beyond gaussian statistical modeling in geophysical data assimilation, *Monthly Weather Rev.*, 138(8):2997–3023, 2010.
16. Otto, M., Germer, T., and Theisel, H., Uncertain 2D vector field topology, *Compu. Graphics Forum*, 29(2):347–356, 2010.
17. Otto, M., Germer, T., and Theisel, H., Uncertain topology of 3D vector fields, *Pacific Visualiz. Symp.*, pp. 67–74, 2011.
18. Petz, C., Pöthkow, K., and Hege, H.-C., Probabilistic local features in uncertain vector fields with spatial correlation, *Comput. Graphics Forum*, 31(3):1045–1054, 2012.
19. Brecheisen, R., Visualization of uncertainty in fiber tracking based on diffusion tensor imaging, PhD thesis, Technische Universiteit Eindhoven, 2012.
20. Jiao, F., Phillips, J., Gur, Y., and Johnson, C., Uncertainty visualization in HARDI based on ensembles of ODFs, In *Proceedings of the 5th IEEE Pacific Visualization Symposium*, pp. 193–200, 2012.
21. Slingsby, A., Strachan, J., Vidale, P., and Dykes, J., Discovery exhibition: Making hurricane track data accessible, http://www.discoveryexhibition.org/uploads/Entries/Slingsby_2010_HurricaneTrackData.pdf, 2010.
22. Potter, K., Wilson, A., Bremer, P., Williams, D., Doutriaux, C., Pascucci, V., and Johnson, C., Ensemble-Vis: A framework for the statistical visualization of ensemble data, In *IEEE Workshop on Knowledge Discovery from Climate Data: Prediction, Extremes.*, pp. 233–240, 2009.
23. Sanyal, J., Zhang, S., Dyer, J., Mercer, A., Amburn, P., and Moorhead, R., Noodles: A tool for visualization of numerical weather model ensemble uncertainty, *IEEE Trans. Visualiz. Comput. Graphics*, 16(6):1421–1430, 2010.
24. Potter, K., Kirby, R. M., Xiu, D., and Johnson, C. R., Interactive visualization of probability and cumulative density function, *Int. J. Uncertainty Quantif.*, 2(4):397–412, 2012.
25. Phadke, M., Pinto, L., Alabi, O., Harter, J., Taylor, R., Wu, X., Petersen, H., Bass, S., and Healy, C., Exploring ensemble visualization, *Visualiz. Tech. Appl.*, 8294:82940B-1–82940B-12, 2012.

26. Martin, J., Swan II, E., Moorehead, R., Liu, Z., and Cai, S., Results of a user study on 2D hurricane visualization, *Eurographics/IEEE Symp. Visualization*, 27(3):991–998, 2008.
27. Broad, K., Leiserowitz, J., Weinkle, J., and Steketee, M., Misinterpretations of the cone of uncertainty in Florida during the 2004 hurricane season, *Am. Meterological Soc.*, 88:651–667, 2007.
28. Bonneel, N., Van De Panne, M., Paris, S., and Heidrich, W., Displacement interpolation using lagrangian mass transport, *ACM Trans. Graphics*, 30(6):158, 2011.
29. Silverman, B. W., *Density Estimation for Statistics and Data Analysis*, Vol. 26, CRC Press, Boca Raton, FL, 1986.
30. Jones, E., Oliphant, T., and Peterson, P., SciPy: Open source scientific tools for Python, 2001.
31. Zwillinger, D. and Kokoska, S., *CRC Standard Probability and Statistics Tables and Formulae*, CRC Press, Boca Raton, FL, 2010.
32. Rubner, Y., Tomasi, C., and Guibas, L. J., A metric for distributions with applications to image databases, In *Computer Vision, 6th Int. Conf.*, IEEE, pp. 59–66, 1998.
33. Bradski, G., *Dr. Dobb's Journal of Software Tools*, 2000.
34. Scott, D. W., Feasibility of multivariate density estimates, *Biometrika*, 78(1):197–205, 1991.