Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2003)
G.-P. Bonneau, S. Hahmann, C. D. Hansen (Editors)

Visualizing Spatial Distribution Data Sets

Alison Luo[1], David Kao[2] and Alex Pang[1]

[1]Computer Science Department, UCSC
[2]NASA Ames Research Center
{alison,pang}@soe.ucsc.edu, davidkao@nas.nasa.gov

Abstract

In this paper, we define distributions as a new data type and address the challenges of visualizing spatial distribution data sets. Numerous visualization techniques exist today for dealing with scalar data. That is, there is a scalar value at each spatial location, which may also be changing over time. Likewise, techniques exist for dealing with vector, tensor and multivariate data sets. However, there is currently no systematic way of dealing with distribution data where there is a collection of values for the same variable at every location and time. Distribution data is increasingly becoming more common as computers and sensor technologies continue to improve. They have also been used in a number of fields ranging from agriculture, engineering design and manufacturing to weather forecasting. Rather than developing specialized visualization techniques for dealing with distribution data, the approach presented in this paper is to find a systematic way of extending existing visualization methods to handle this new data type. For example, we would like to be able to generate isosurfaces of 3D scalar distribution data sets, or generate streamlines of vector distribution data sets. In order to accomplish this goal, we propose the use of a set of mathematically and procedurally defined operators that allow us to work directly on distributions. Color images can also be found in www.cse.ucsc.edu/research/avis/operator.html.

1. INTRODUCTION

A host of visualization techniques is available today to visualize a variety of data types. The data sets may be characterized as both multidimensional and multivariate. Multidimensional data refers to the spatial dimensionality e.g. 0D, 1D, 2D, 3D, of the data, but it may also include time as an additional dimension. Multivariate data, on the other hand, refers to the different variables represented at each location. These variables are usually scalar, but may also be vectors, tensors, etc. For non-scalars, one may treat the extra terms as another variable in much the same way that vectors may be represented by multiple scalar components. While often used interchangeably in literature, these two properties are orthogonal. For example, a weather forecast may be 3D, time varying and contain information about temperature, humidity, pressure, etc. at each location. In practice, such a data set may be stored in a 5D array: three for space, one for time, and the last one for the different variables. A notable visualization system that carries this name is Vis5D [7].

This paper focuses on a new data type, a distribution, which will essentially add an extra dimension that needs to be visualized. A distribution is simply a collection of n values about a single variable: $D = v_i$, where $i = 1..n$. For example, a probability density function (pdf) is a distribution containing values that represent frequencies of different data values. In terms of visualizing distributions, we are all familiar with bar plots which characterize the statistical distribution for a single variable at a given location. We are less familiar with distributions mapped over one, two or three spatial dimensions. So, the goal of this paper is to describe a systematic methodology for visualizing distribution data that are 2D or higher.

Distribution data are distinctly different from multivariate data in that one can have scalar distribution data where each member of the collection is a scalar, one can have vector distribution data where each member of the collection is a vector, or one can have multivariate distribution data where each member of the collection is a multivariate vector. Alternatively, one can think of each component of a non-scalar distribution as separate scalar distributions (see Figure 1).

Rather than developing specialized visualization techniques for visualizing spatial distribution data sets, the ap-

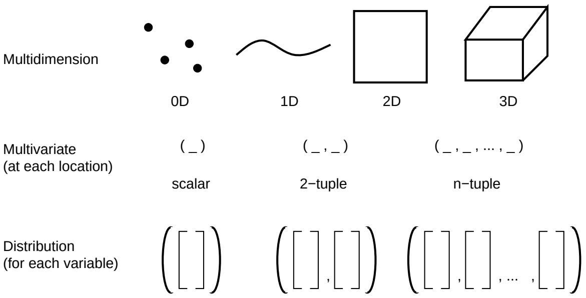


Figure 1: *Multidimensional, time-varying, multivariate distributions.*

proach presented in this paper is on a systematic methodology of extending existing visualization techniques to handle the new data type. Thus, one would like to be able to pseudocolor a distribution field, or generate contour lines or isosurfaces, or streamlines or volume renderings of distribution fields. To accomplish this, we describe a set of operators that converts distributions to scalars, combines distributions arithmetically, and possibly more complex operations on distributions. Because distributions do not always represent statistical information, different types of operators may be called for depending on what is appropriate for the application that generated distribution data. We discuss a few of these operators from various fields including: signal processing, information theory, and statistics. The intent is to demonstrate how one may go about designing operators rather than providing an exhaustive list.

The rest of the paper is organized as follows: Section 2 provides additional motivation on why it is important to be able to deal with distribution data sets, Section 3 highlights three distribution data sets that are used to demonstrate the visualization techniques in this paper and Section 4 presents extensions to a number of well known visualization techniques to support distribution data sets. For each visualization technique, we present and discuss the relevant operators necessary to handle distribution data.

2. MOTIVATION

Distribution data can be found in raw census data, real estate sales data, agriculture, bioinformatics, sensitivity analyses, terrestrial models, weather forecasts, and ocean circulation models to name a few. More specific examples include: modeling of vegetation and land cover types using conditional simulation [3], data assimilation into ocean circulation models [12], target state estimation using Bayesian techniques [16], studying gene sequences and gene expression levels from micro-arrays [4], and to some extent, query by image features [6] and music content [13, 17] into digital libraries. These applications all provide a rich set of data.

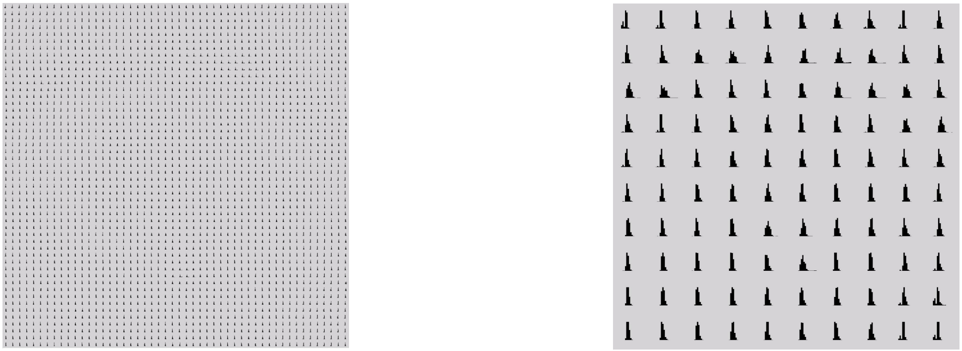
Another reason why distribution data is interesting is because it is another way to represent uncertainty. Rather than using a scalar value or value pairs such as standard deviation or min-max range values to represent uncertainty, the entire distribution itself can be the representation of uncertainty.

Despite all the potential benefits of distribution data, their prevalence is not immediately obvious. We believe the primary reason is due to the fact that they are difficult to visualize when they are spread out over a field. Currently, spatial distribution data are generally summarized using a few aggregate statistics and presented that way because there is just no visualization method currently available that will depict spatial distribution data. Such methods of displaying these data also hides the fact that they are distributions.

If the distribution data exists only at a single “point”, then the visualization is relatively straightforward by using function plots or bar charts. However, as the spatial dimension of the data set increase from a point to 1D, 2D, 3D and time-varying, then the visualization task very quickly becomes a problem.

We have looked at this problem for 2D distribution data sets using a statistical approach [10]. Density estimates were constructed for the distribution at each pixel over a 2D domain. From that, parametric statistics such as mean and standard deviations, as well as higher moment statistics were collected. These were then displayed using different visual mappings such as color, height, glyphs, etc. over the domain. This approach works well if the distributions can be characterized fairly well with a few statistical parameters e.g. if the distributions can be well modeled by a Gaussian distribution. However, as the modality or the number of peaks in the distribution deviate from the norm, then the parametric statistics approach has a severe limitation in expressing the shape of the distributions. An alternative method proposed was to treat the density estimates over the 2D field as a 3D scalar volume (2D for space, 3rd dimension for data range, and scalar values representing frequencies) and apply standard volume visualization algorithm. This worked better but doesn’t allow us to scale to higher dimensional distribution data sets.

Employing multivariate visualization techniques on the 2D distribution data met with very limited success. A case in point is the use of glyphs e.g. Chernoff faces or star glyphs to represent multivariate information. The most natural glyph to represent distributions would be bar charts. Figure 2 illustrates how this looks like for 2D distribution data (described in Section 3 as `sg2`). One obvious limitation is the difficulty of scaling this approach to high resolution data sets. A relatively low resolution 100x100 grid of distribution can barely be handled by this approach, even when it is displayed at quarter resolution. The lower resolutions are derived by aggregating the samples of neighboring points together to form a new distribution. While preserving the raw data points, one also loses the spatial variability between neighboring distri-



(a) quarter resolution

(b) 1/100 resolution

Figure 2: *Bar chart glyphs representing distributions are drawn over each distribution. (a) shows the original 100 x 100 grid displayed as a 50 x 50 array of histograms, while (b) shows the same data as a 10 x 10 array of histograms.*

butions. Another limitation to this and other multivariate approach is the difficulty in scaling to higher dimensions.

In short, there are clearly many applications with distribution data. The most common practice has been to summarize the distribution data and work with a scalar summary. Analyzing and visualizing the distribution data in its entirety is the challenge that this paper tries to address. In the next section, we present some distribution data sets used in this paper.

3. DATA FOR EXPERIMENT

We describe three different distribution data sets used in this paper. Our first data set is a 2D scalar distribution field. This data set is a synthetic example constructed using a small region in the Netherlands imaged by the Landsat Thematic Mapper [3]. Consider the case in which the biophysical variable to be mapped across this region is percent forest cover. Assume there are ground-based measurements of forest cover from 150 well-distributed locations throughout this region as well as space-based measurements from Landsat of a spectral vegetation index. This spectral vegetation index is related to forest cover in a linear fashion but with significant unexplained variance. Further assume that the ground area represented by a field measurement is equal to the area represented by one pixel. A distribution data set was generated using this information: `sg2`, generated using a conditional simulation algorithm [2] taking into account ground measurements only; and `sg3` generated using conditional co-simulation [2] using both ground measurements and the coincident satellite image. The data set consists of 101 × 101 pixels and 250 realizations. Values range from 0 to 255, rescaled from percent forest cover.

Our second data set is a 3D time-varying scalar distribution field output from ocean modeling. The model covers the Middle Atlantic Bight shelfbreak which is about 100 km wide and extends from Cape Hatteras to Canada. Both measurement data and ocean dynamics are combined to produce a 4D field that contains a time evolution of a 3D volume including variables such as temperature, salinity, and sound speed. To dynamically evolve the physical uncertainty, an Error Subspace Statistical Estimation (ESSE) scheme [12] is employed. This scheme is based on a reduction of the evolving error statistics to their dominant components or subspace. To account for nonlinearities, they are represented by an ensemble of Monte-Carlo forecasts. Hence, numerous 4D forecasts are generated and collected into a 5D field. For each physical variable, the dimension of the data set is 65 × 72 × 42 voxels with 80 values at each point. We refer to this data set as `ocean`.

Our third data set is a 3D time-varying multivariate distribution data set representing an ensemble weather forecast. The data set is referred to as `sref` which stands for short-range ensemble forecasting and is courtesy of NOAA, and available through http://wwwt.emc.ncep.noaa.gov/mmb/SREF/SREF.html. The ensemble is created from two different models: ETA and RSM, with 5 different initial and boundary conditions each producing an ensemble or collection of 10 members at each location where the two models overlap. Unfortunately, the two models are not co-registered and have different projections and spatial resolutions. Thus, for the purpose of this paper, we just use the five member ensemble from the RSM model. The resolution of the RSM model is 185 × 129 and has 254 physical variables at each location. The forecast is run twice a day, and for 22 different time steps during each run. Velocity is available at every location in the model. However, only horizontal wind components are recorded. While not an ideal distribution data set because of the low number of samples, it provides us with velocity distribution data to demonstrate streamlines and pathlines visualization of steady and unsteady vector distribution data.

4. VISUALIZATION AND OPERATORS

We now describe how to extend five standard visualization to support distribution data sets. These are pseudocoloring, streamlines, pathlines, contour lines and isosurfaces. With each method, we discuss the basic requirements in order to extend it, and also present different ways of extending it. In all cases, the key idea is designing operators that can convert distribution data types to other types and/or allow them to be combined with other data types. Based on these examples, a methodology is established for how other visualization methods can be extended to support distribution data sets.

4.1. Pseudocoloring

One of the most basic visualization techniques is pseudocoloring. The main requirement is to map a scalar value to a color value. Typically a range of scalar values are mapped to a range of color values. Disregarding the importance of designing a proper colormap for now, the key task in order to pseudocolor distributions is to convert a distribution to a scalar value. Assuming an operator exists for this task, the scalar value representing the distribution can then be used to index the color table. Note that the mapping need not be linear. It could be nonlinear or even discontinuous. Alternatively, a distribution can be converted to a few scalars, say three, rather than a single scalar. In this case, the three scalars can be mapped to different color models such as HSV components. This class of operators converts a distribution to a single scalar or a vector of a few scalar components.

$$s = ToScalar(D) \tag{1}$$

$$v = ToVector(D) \tag{2}$$

In choosing an operator to convert a distribution to a scalar, it is important to know what features of the distribution need to be shown. We describe some statistical operators that summarize a distribution to a few scalars. The central tendencies of a distribution often show what the most likely values are. The spread of a distribution captures variability or uncertainty. Additional measures such as kurtosis describes the “flatness" of the distribution and skewness describes the asymmetry. The equations for these statistical summaries can be found in statistics textbooks [9]. In addition to standard statistical summaries, one can also devise other more descriptive means of expressing the shape of a distribution. For example, capturing the modality or number of peaks in a distribution, and the height and width of each peak [11]. Figures 3, 4 and 5 demonstrate how these ToScalar() and ToVector() operators are used over 2D distribution data sets. It should also be noted that these examples are illustrative and are by no means exhaustive.

4.2. Streamlines

Streamlines is one of the workhorse visualization techniques for steady state flow fields. They are generated by integrating the path of massless particles as they are carried instantaneously through the field. For illustration purposes, we use the simplest Euler integration outlined below:

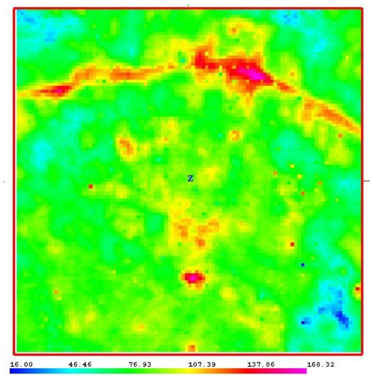


Figure 3: *Pseudocolor rendering of the mean of the* `sg2` *data. Mean is obtained using a ToScalar() operator defined in Equation 8.*

$$P_{i+1} = P_i + \vec{v}\Delta t \tag{3}$$

Where P is the position, $\vec{v}$ is the velocity and Δt is the integration step. If the velocity field is a velocity distribution field, we need to extend the concept of multiplying and adding scalars with distributions. The result of multiplying a scalar with a distribution is another distribution where each term or sample has been scaled.

$$D' = Scale(s, D) = s \times D \tag{4}$$

Note that this is carried out for each velocity component. Here, D' is the new distribution for one of the velocity components. Likewise, adding a scalar to a distribution simply offsets the distribution by the scalar amount.

$$D' = Add(s, D) = s + D \tag{5}$$

Again, note that each component of the position P is added to the corresponding distribution component. After one integration step of Equation 3, the right hand side is now a distribution. So, we need to apply one of the ToScalar() operators described earlier to each of the components of P in order for us to assign the results to the left hand side.

In the example below, we use a 2D slice from the `sref` forecast data. There are five velocities at each location in the 2D field. Each velocity has two components, u and v. We show the results using three different ToScalar() operators: mean, minimum, and maximum. Note that these operators are applied to the distribution of new positions (the right hand side).

$$Minimum(D) = min(v_1, v_2, ..., v_n) \tag{6}$$

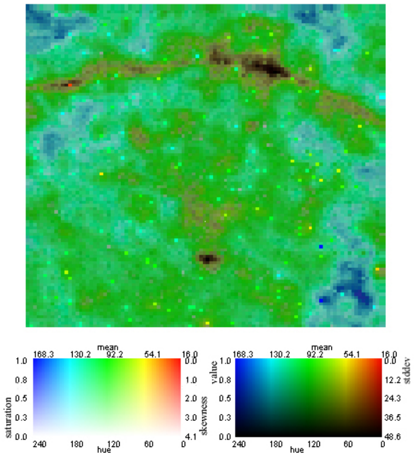


Figure 4: *Pseudocolor rendering of the three parameters from* `sg2` *obtained using a ToVector() operation that extracted the mean, standard deviation, and skewness of the distribution. Mean is mapped to hue using the same color range as in Figure 3, standard deviation is inversely mapped to value, and the absolute value of skewness is inversely mapped to saturation. The locations of the ground truth points are also easily visible as brighter, fully saturated points. Places with higher standard deviations are showing up as darker regions, especially prominent across the arch and the lower, middle region. The color map on the left has the value held constant at one, while the color map on the right has the saturation also held constant at one.*

$$Maximum(D) = max(v_1, v_2, ..., v_n) \tag{7}$$

$$Mean(D) = \frac{1}{n}\sum_{i=1}^{i=n} v_i \tag{8}$$

Figures 6 and 7 represent one possible interpretation of the vector distribution data. Each realization is essentially treated as a possible scenario and hence a streamline is generated independently from each realization. Another possible interpretation is to take all the different realizations together and treat the vector distribution at each point as a pdf of the velocity at that point. Given that five samples is generally insufficient for an accurate pdf, we continue with the following discussion to illustrate how one might proceed with such an interpretation assuming an adequate population is available.

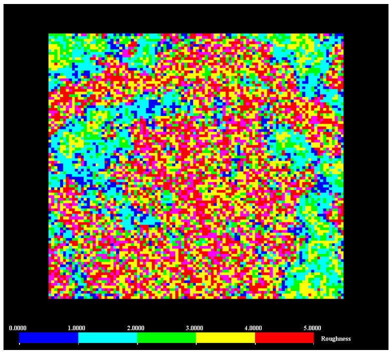


Figure 5: *Discontinuous color map of output from a bump hunting algorithm on the* `sg2` *data set. The number of bumps (or modality) of a distribution is mapped to different colors. The output of the bump hunting algorithm* [11] *presented here is a procedural ToScalar() operation. The arch observed in Figured 4 is also noticeable here. The reddish region indicates that the distributions at those locations are more bumpy (higher modality).*

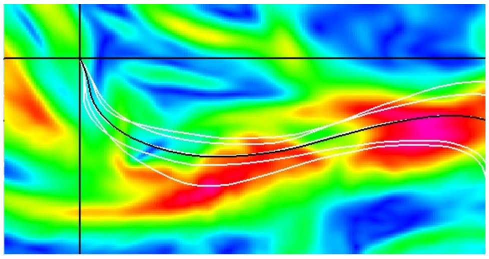


Figure 6: *Streamlines of the* `sref` *RSM models on October the 24th of 2002. The background field is colored with the mean velocity magnitude using the same color map as in Figure 3. The white streamlines are traditional streamlines calculated independently over each vector field realization. One can see why they are referred to as spaghetti plots in meteorology. The black streamline uses Equation 8 to convert the distribution at each component of* P_{i+1} *to a scalar. It also corresponds to what one might expect as the average streamlines of all the spaghetti plots.*

The alternative interpretation basically says that starting from a given position, one has a pdf which specifies where one might end up after one integration step. As a consequence, the new location is going to be a distribution of likely positions. We therefore look at another way of extending Equation 3 to work with distributions by raising the data type of positions from scalars to distributions. The initial seeding point P_0 is determined as usual, but is converted into a distribution by simply replicating it so that it has the same number of samples as the distributions in the velocity distribution field. We then define a different addition operator that works on two distributions:

$$R = AddD(P, Q) = P \bigoplus Q \tag{9}$$

The result of AddD() is another distribution R. Again, there is more than one way to define such an operation. We de-

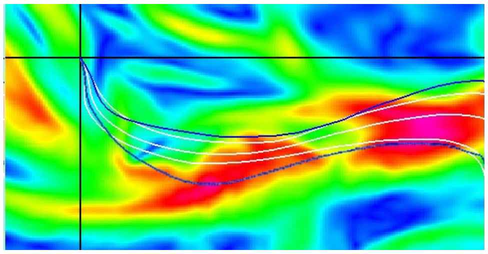


Figure 7: *Same as Figure 6 but showing the spaghetti plots with the streamlines generated using Equations 6 and 7. The two blue streamlines correspond to the envelope of the spaghetti plots.*

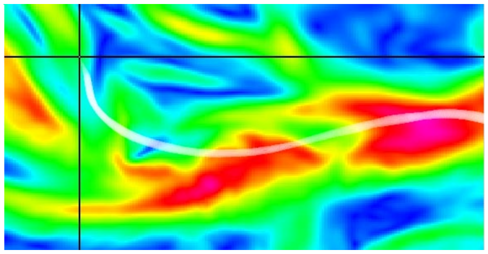


Figure 8: *Streamline visualization using the convolution addition operation. The distribution of positions at the end of each time step is rendered as a transparent circle encompassing those points.*

scribe two possible definitions of $\bigoplus$. The first is due to Gerasimov et al. [5] which we refer to as "convolution addition", and the second one is due to Gupta and Santini [6] which we refer to as "binwise addition". Figures 8 and 9 demonstrate these two operations on the ensemble weather forecast data set.

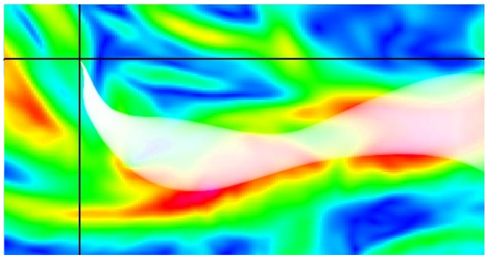


Figure 9: *Streamline visualization using the binwise addition operation. It is rendered with transparent circles as in Figure 8, and has the same seed point as the previous three streamline images.*

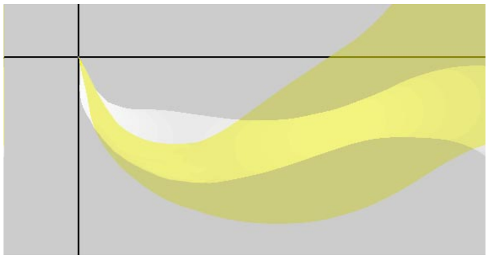


Figure 10: *The white swath represents streamline trajectories, while the yellow swath represents pathline trajectories of the ensemble. Both swaths use binwise addition.*

4.2.1. Convolution addition:

This addition is statistically meaningful when P and Q are pdfs. Let P be the pdf of random variable X and Q be the pdf of random variable Y. The addition of these 2 independent pdfs results in another pdf which can be interpreted as the probability distribution for the sum of both random variables. The relationship is defined in [5] as:

$$P_{X+Y}(t) = \int_{-\infty}^{+\infty} P(t-v)Q(v)dv \qquad (10)$$

where $t = X+Y$ and $v = Y$.

4.2.2. Binwise addition:

This addition does not require that the distributions be pdfs, but do require that both be evaluated over the same range of X values. At each evaluation point, the frequencies of both variables are simply added up. If the distribution is a histogram, then corresponding bins are added together to form the frequency of bin in the resulting distribution. This is defined in [6] as:

$$R(X) \;=\; P(X) \;+\; Q(X) \qquad (11)$$

Another application where such operations can be seen is in signal processing. A composite signal $R(t)$ of two signals $P(t)$ and $Q(t)$ can be obtained from the above operation. In this case, $t \;=\; X$.

4.3. Pathlines

Pathlines trace the path of a particle over a time varying flow field [1]. A particle's path is determined by solving the differential equation for a field line:

$$dP/dt \;=\; V(P(t),t) \qquad (12)$$

where P is the particle's location and V the particle's velocity at time t. Integrating this equations yields

$$P(t+dt) \;=\; P(t) \;+\; \int_{t}^{dt} V(P(t),t)dt \qquad (13)$$

The integral can be evaluated numerically using a simple Euler integration, multi-stage methods (e.g. Runge-Kutta, Bulirsch-Stoer), or multi-step methods (e.g, backwards differentiation). Note that this equation is essentially the same as Equation 3 with the exception of a time varying V field.

Pathlines from ensemble time varying flow data are illustrated in Figure 10. Note that pathlines can be rendered with polylines or as transparent circles as with streamlines. Recall that the `sref` data has 22 time steps for each forecast run. These time steps are three hours apart and a bit too large of a time step for integration purposes. Using a single forecast run, we need to do temporal interpolation of the ensemble flow field. For the example in Figure 10, we introduce 30 additional time frames between each three hour time step using the interpolation strategy for distributions described in Equation 19. This allows an integration step size of 0.1 hours.

4.4. Contour Lines

An isolevel or contour line is a curve linking all the points with the same value. On a gridded data set, bilinear interpolation is usually performed to obtain a better approximation to where the curve will intersect a cell based on the relationship between the corner values and the reference value for the contour. In the same vein, we would like a contour line over a 2D distribution field to link all the points in the field with the same distributions.

There are two ways to do this. The first approach is to convert each distribution to a scalar value using one of the ToScalar() operators described earlier, and then apply a scalar contouring algorithm. The second approach is to contour the 2D distribution field directly. The latter requires two basic changes. First, the reference scalar value for locating the contour line must be extended to become a reference distribution. That is every distribution in the field will be compared against this reference distribution, just as every scalar in a 2D scalar field is compared with the reference scalar. Secondly, there must be a way to measure similarity between two distributions. This allows us to compute the distances between each distribution in the 2D grid against the reference distribution. We call this class of distance measures our similarity operators.

$$s \;=\; Similarity(D_1, D_2) \tag{14}$$

Again, we provide a non-exhaustive list of illustrative examples of how this class of operators may look. Here, we introduce three such operators: Euclidean distance operator, Kolmogorov-Smirnov operator and Kullback-Leibler operator. They measure the similarities or differences between two distributions. Let $P(x)$ and $Q(x)$ be two distributions of the random variable x. The Euclidean distance is defined as:

$$Euclidean(P,Q) \;=\; \left(\int_{-\infty}^{+\infty} (P(x)-Q(x))^2 dx\right)^{\frac{1}{2}} \tag{15}$$

The $P(x)$ and $Q(x)$ may be evaluated at discrete locations e.g. from a histogram, or at continuous locations from a density estimate [11, 15].

The Kolmogorov-Smirnov (KS) distance between two distributions is defined as the maximum distance between their corresponding cumulative distribution functions (cdf) [18].

$$KS(P,Q) \;=\; \max|cdf(P(x)) - cdf(Q(x))| \tag{16}$$

We can also find a distance measure in information theory. This is the Kullback-Leibler (KL) distance also called relative entropy, and is defined as :

$$KL(P,Q) \;=\; \int_{-\infty}^{+\infty} P(x) \log \frac{P(x)}{Q(x)} dx \tag{17}$$

Here, P and Q are treated as pdfs. From the formulation, we note that the KL distance is always non-negative. When the two pdfs are identical, their KL distance is zero. The greater the KL distance, the bigger the difference between the two pdfs. The conventional KL distance becomes infinity when $Q(x)$ is zero. To avoid this problem, an alternative formulation proposed in [8] is to add small non-zero values to the pdfs. The KL distance is also not symmetric. That is, $KL(P,Q) \neq KL(Q,P)$. Siegler *et al.* [14] proposed a symmetric KL (SKL) distance which is defined as:

$$SKL(P,Q) \;=\; KL(P,Q) + KL(Q,P) \tag{18}$$

In the following examples, We combine these two solutions, non-zero padding and symmetrization, to produce the SKLZ distance.

We use the marching squares contouring algorithm where each vertex distribution is compared against the reference distribution. Based on the results of that comparison, a determination is made whether a contour line will cross an edge or not. If a line must cross an edge, we still need to determine the intersection. Linear interpolation does not work here, because the distance does not change linearly between any two given points in 2D space. So we subdivide the edge into 10 sections and find the section whose distribution is closest to the reference distribution. That is, we generate 9 intermediate distributions that are linear combinations of the distributions at the ends of the edge to be intersected using:

$$Interp(P,Q,s) \;=\; Scale(1-s,P) \bigoplus Scale(s,Q) \tag{19}$$

where Scale() is defined in Equation 4 and $\bigoplus$ is the binwise addition described in Equation 11.

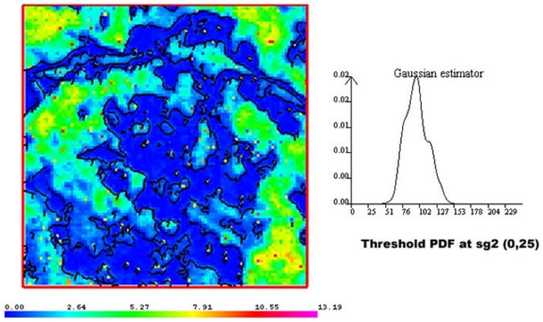


Figure 11: *Contours of distribution data sg2 using SKLZ operator. The distribution to the right is the reference distribution used to find the contour lines. That is, points along the contour lines have distributions very similar to the reference distribution according to the SKLZ similarity measure. Color corresponds to the output of the SKLZ similarity measure.*

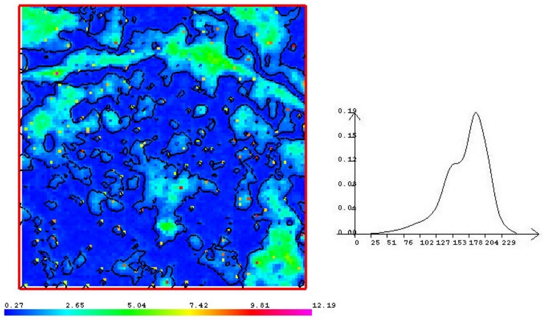


Figure 12: *Similar to Figure 11 except a different reference distribution is used. The reference distribution is made up by aggregating all the samples of all the pixels in the data set. The dots correspond to locations of the ground truth points. Color corresponds to the output of the SKLZ similarity measure.*

4.5. Isosurfaces

Extending isosurfaces to deal with distribution data is a straightforward extension of how contour lines were extended. We illustrate how this can be done using the 3D distribution of sound speed from the `ocean` data set. Finding an accurate intersection of a polygon and an edge of a volume cell can be quite costly because using the strategy outlined in the previous section, we have to effectively increase the spatial resolution by three orders of magnitude. Therefore, we forego finding accurate polygon-edge intersections in the following example. We still use a Similarity() operator such as the SKLZ distance measure to compare the distribution at each vertex against the reference distribution. This produces a scalar value that we then pass to a standard marching cubes algorithm.

5. EVALUATION OF OPERATORS

In the preceding section, we described how existing visualization algorithms can be extended to handle distribution data sets using different types of operators. Within a given type of operator, such as ToScalar(), Similarity(), etc., there is usually more than one possible operator definition. It is therefore natural to ask how to select the right operator. The choice of the right operator clearly depends on the application at hand, and possibly the properties of the distribution e.g. number of samples, statistical nature, etc. It also depends on what features in the data, or distribution data, are being highlighted. Discussion on the selection procedure among choices of operators for a specific application is beyond the scope of this paper. Instead, we discuss one generic method that compares alternative operators. The assumption is that the user has identified meaningful properties and would like to know which operator is best at detecting them. The operators are evaluated using power tests. The power of a statistical test is the probability of rejecting the null hypothesis when it is false. In other words, it is the probability that the test will do what it was designed to do. To illustrate how the power test works, assume that the user has selected three operators to evaluate: the Kullback Leibler (KL), Kolmogorov-Smirnov (KS) and an Euclidean distance operator (ED). Further assume that the distribution property of interest is skewness. Thus, we would like to evaluate these three operators with regard to their sensitivity to changes in skewness. For a controlled test, we create different pdfs from a Chi-square distribution with different shape parameters.To further isolate the differences among these pdfs to skewness, the distributions are transformed so that they all have the same mean of 0 and standard deviation of 1. These are used as input to the power test.

Given two distributions P and Q, the null hypothesis in this context is true when the two distributions are the same. The alternative hypothesis is true when they are different. Each pdf derived from a χ^2 distribution is compared against the pdf derived from a normal distribution with zero skewness. We carry out the test as follows:

1. Evaluate each of the χ^2 pdfs at 250 equally spaced points. Transform each one such that the mean is 0 and the standard deviation is 1.
2. Construct a cdf for each of these.
3. For each cdf, obtain 2,500 samples through uniform sampling. This produces a fairly accurate reconstruction of the input distribution. Now, generate 100 such reconstructions for each cdf.
4. Calculate distances (using KL, KS, ED) between distributions from the same input distribution (same

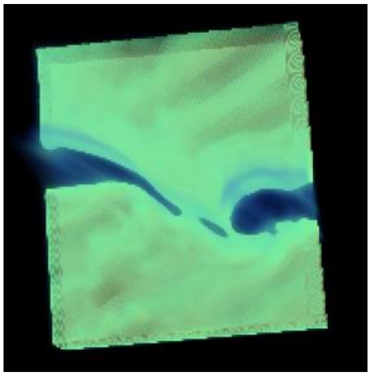


Figure 13: *Volume rendering of the* `ocean data` *after a ToScalar() operation to obtain the standard deviation of the distributions. The darker region lies above the continental shelf break where more mixing is happening, thus, the higher standard deviations in the region.*

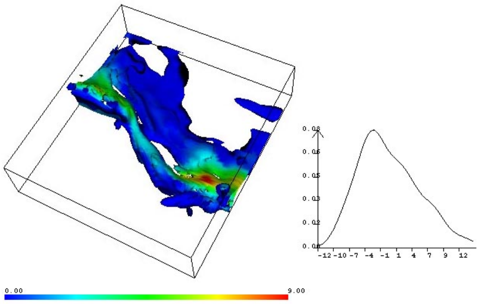


Figure 14: *An isosurface using a reference distribution within the mixing region. The surface represents regions in the data where the distributions are very similar to the reference distribution shown on the right. Not surprisingly, it corresponds quite well with Figure 13. Color can also be used to display other properties of the distributions by using an appropriate ToScalar() operator. Here we color the surface by the standard deviation of the distribution at each location.*

skewness). Here the null hypothesis is true. Each comparison produces a scalar distance measure, where the distances should be relatively small.

5. Calculate distances (using KL, KS, ED) between each skewed distribution and the normal distribution. Here the alternative hypothesis is true. Each comparison produces a scalar distance measure, where the distances should be relatively larger than those in the null hypothesis.
6. Plot the two groups of distances to produce Figure 15. A plot like this is generated for a given Similarity() operator and one skewed distribution. The area under the black curve and to the right of the yellow line is the discriminating power, and provides a single data point for the plot in Figure 16.

The following observations can be made from Figure 16. First, the KL operator is the most powerful distance measure among the three operators for detecting differences in skewness. It is followed closely by the ED operator. The performance of the KS operator is not monotonically improving with skewness. Secondly, as the skewness increases, the Chi-square distributions become more and more distinguishable from the normal distribution. Therefore, the power of the test goes up monotonically, except for KS operator.

The power test can also be applied to arbitrary distributions such as those from the three test data used in this paper. However, the test is quite costly, and it does not replace the knowledge and experience of the user in deciding whether a particular operator is meaningful for the desired task or not.

6. CONCLUSION

We presented a methodology for visualizing spatial distribution data sets. It is based on operators and we demonstrated how it is used to extend some basic visualization techniques to handle distribution data sets. The methodology is flexible and can grow by allowing one to take advantage of developments in other domains such as signal processing, statistics, etc. The flexibility comes at a price in terms of requiring care and some learning in proper interpretation of the resulting images. However, it opens up the visualization research field by first adding distributions to the list of data types that we can visualize, and second by allowing us to extend visualization techniques to support distribution data sets, and finally to perhaps find visualization techniques with distribution data foremost in mind. For us, one of the main benefits is that uncertainty represented as a distribution can now be visualized relatively directly with extensions to standard algorithms.

A key point for discussion is that we have introduced a broad set of operators from different fields. Their judicious use requires some knowledge and experience. More importantly, how does one know which operator to use for a given data set and how does one interpret the resulting images? We do not have any guidelines or rule of thumb to offer at the present time, but this is part of our plan to evaluate the different options available and to be able to recommend appropriate ones later on.

Other future work plans include feature extraction of distribution data, specially time-varying distribution data, as this holds great promise in helping to visualize such rich and large data sets. We also plan to explore how other visualization techniques aside from those presented here can be

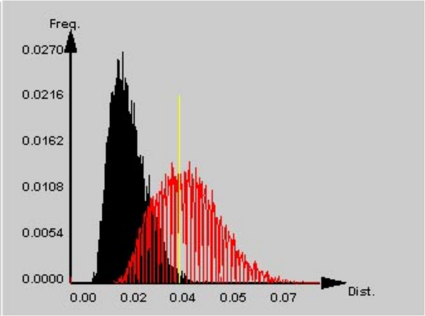


Figure 15: *The black distribution is obtained when the null hypothesis is true and the red distribution is obtained when the alternative hypothesis is true. The x-axis is the distance measure using one of the three operators, while the y-axis is the relative frequency. The yellow line is the marker indicating type I error probability (α) and is set at 1% which corresponds to the area under the black curve to the right side of the marker.*

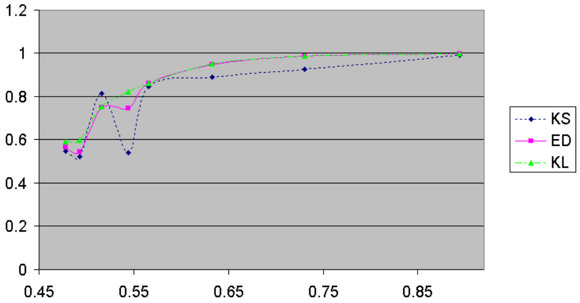


Figure 16: *Comparison of Chi-square distributions with different shape parameters (different amounts of skewness). The higher the curve, the more discriminating (more powerful) the distance measure.*

extended to work with distribution data sets. Finally, we also plan to explore how scattered data interpolation techniques can be extended to handle distribution data sets.

7. ACKNOWLEDGEMENTS

This work is supported in part by the NASA Intelligent Systems (IS) Program under Cooperative Agreement NCC2-1260 and NSF ACI-9908881. We would like to thank Jennifer Dungan, David Draper, David Helmbold, Wendell Nuss and Peter de Souza for technical discussions, Pierre Lermusiaux for providing the ocean data, and Anna Chen, Newton Der, Jose Renteria, Wei Shen, and Bing Zhang for their valuable help.

References

1. D.L. Darmofal and R. Haimes. An analysis of 3D particle path integration algorithms. *Journal of Computational Physics*, 123(1):182–195, January 1996.
2. C.V. Deutsch and A. G. Journel. *GSLIB: Geostatistical Software Library*. Oxford University Press, New York, 1998.
3. J. L. Dungan. Conditional simulation: An alternative to estimation for achieving mapping objectives. In F. van der Meer A. Stein and B. Gorte, editors, *Spatial Statistics for Remote Sensing*, pages 135–152. Kluwer, Dordrecht, 1999.
4. DJ. Lockhart et. al. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, 14:1675–1680, 1996.
5. V.A. Gerasimov, B.S. Dobronets, and M.Yu. Shustrov. Numerical operations of histogram arithmetic and their applications. *Automation and Remote Control*, 52(2):208–12, Feb 1991.
6. Amarnath Gupta and Simone Santini. Toward feature algebras in visual databases: The case for a histogram algebra. In *Advances in Visual Information Management. Visual Database Systems*, pages 177–196, 2000.
7. William Hibbard and David Santek. The Vis-5D system for easy interactive visualization. In *Visualization '90*, pages 28–35, 1990.
8. Don H. Johnson. A theory of information processing. www.ima.umn.edu/talks/workshops/1-29-2-2.2001/johnson/johnson.pdf.
9. Robert Johnson and Patricia Kuby. *Elementary Statistics*. Duxbury, 2000.
10. David Kao, Jennifer Dungan, and Alex Pang. Visualizing 2D probability distributions from EOS satellite image-derived data sets: A case study. In *Proceedings of Visualization '01*, pages 457–460, 2001.
11. David Kao, Alison Luo, Jennifer Dungan, and Alex Pang. Visualizing spatially varying distribution data. In *Proceedings of the 6th International Conference on Information Visualization '02*, pages 219–225. IEEE Computer Society, 2002.
12. P.F.J. Lermusiaux. Data assimilation via error subspace statistical estimation, Part II: Middle Atlantic Bight shelfbreak front simulations and ESSE validation. *Monthly Weather Review*, 127(7):1408–1432, 1999.
13. Beth Logan and Ariel Salomon. A music similarity function based on signal analysis. In *IEEE International Conference on Multimedia and Expo (ICME)*, August 2001.
14. Matthew A. Siegler, Uday Jain, Bhiksha Raj, and Richard M. Stern. Automatic segmentation, classification and clustering of broadcast news audio. In *DARPA Speech Recognition Workshop*, pages 97–99, Chantilly, Feb. 1997.
15. B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall, 1986.

16. Lawrence D. Stone, Carl A. Barlow, and Thomas L. Corwin. *Bayesian Multiple Target Tracking*. Artech House Radar Library, 1999. 300pp.

17. M. Welsh, N. Borisov, J. Hill, R. von Behren, and A. Woo. Querying large collections of music for similarity. Technical Report UCB/CSD00 -1096, U.C. Berkeley Computer Science Division, November 1999.

18. W.T.Eadie. *Statistical methods in experimental physics*. Amsterdam, North-Holland Pub. Co., 1971.