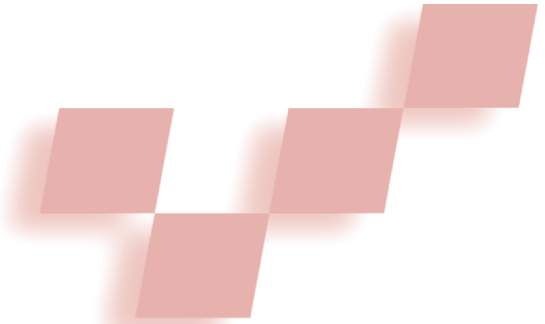


Visualizing Spatial Multivalue Data

Alison L. Love and Alex Pang
University of California, Santa Cruz

David L. Kao
NASA Ames Research Center

We introduce multivalue data as a new data type in the context of scientific visualization. While this data type has existed in other fields, the visualization community has largely ignored it. Formally, a multivalue datum is a collection of values about a single variable. This collection can be denoted as $M = (v_1, v_2, \ldots, v_n)$ where each v_i is a value of variable v. The collection might arise from a measurement process or a modeled process. In the latter case, it is useful to consider probabilistic models where the collection of values describes the set of possible outcomes of the modeled process.

Multivalue data sets can be defined for multiple dimensions. A spatial multivalue data set consists of a multivalue datum at each physical location in the domain. The time dimension is equally valid. This leads to spatiotemporal multivalue data sets where there is time varying, multidimensional data with a multivalue datum at each location and time.

To illustrate the challenge of visualizing spatial multivalue data sets, consider having to visualize a time varying volumetric temperature field. We could possibly generate direct volume renderings of each temperature volume and create an animation, or extract temperature isosurfaces and track how they change over time. Either method can help reveal how the temperature volume changes over time. Now, imagine that instead of this time varying volumetric temperature field, we now have n versions of them, each one slightly different from the other. Using the traditional approach, we would have produced n volume rendered animations, or n isosurface animations. The variations, consensus, or other group properties of such a multivalue data set would be difficult to comprehend by sequentially watching these animations.

Compounding matters, the previous example was just a simple scalar multivalue data set. We also need to consider vector, or more generally, multivariate multivalue data sets. Specifically, another data descriptor is whether a data set is univariate or multivariate. A multivariate multivalue data set has a multivalue datum for each variable. Multivariate data are often represented as a vector of values—one for each variable. While the term multidimensional data is often used interchangeably with multivariate data in literature from different disciplines, we want to make a clear distinction between their use in this article. *Multidimensionality* refers to the spatial and temporal dimensions of a data set, while *multivariateness* refers to the number of variables a data set describes. This distinction is important because these are orthogonal concepts, and therefore require different treatment, particularly when it comes to visualizations and their interpretation.

Note that while both multivalue data and multivariate data can be represented as a vector of values, they are conceptually distinct. Hence, they also require different treatment when it comes to visualization. That is, while multivariate visualization techniques might be applied to multivalue data, they are not necessarily appropriate. For example, a common objective with multivariate data visualization is to find relationships among variables. On the other hand, finding relationships among different instances of the same variable is not usually relevant.

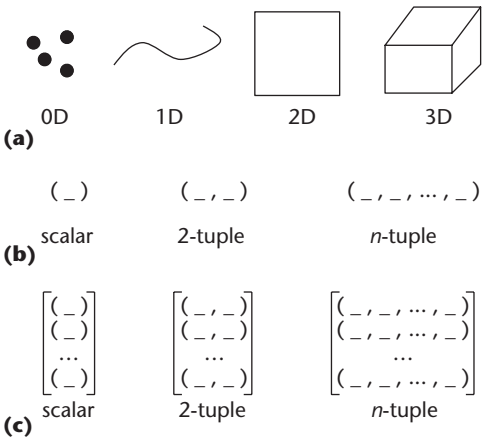
The spatial multivalue data type captures multiple instances of the same variable at each location in space. Visualizing spatial multivalue data sets is a new challenge.

To recapitulate, we have three orthogonal data descriptors: multivalue, multivariate, and multidimensional. A complex data set can have all three properties leading to a multidimensional, multivariate, multivalue data set. Figure 1 (next page) illustrates these concepts .

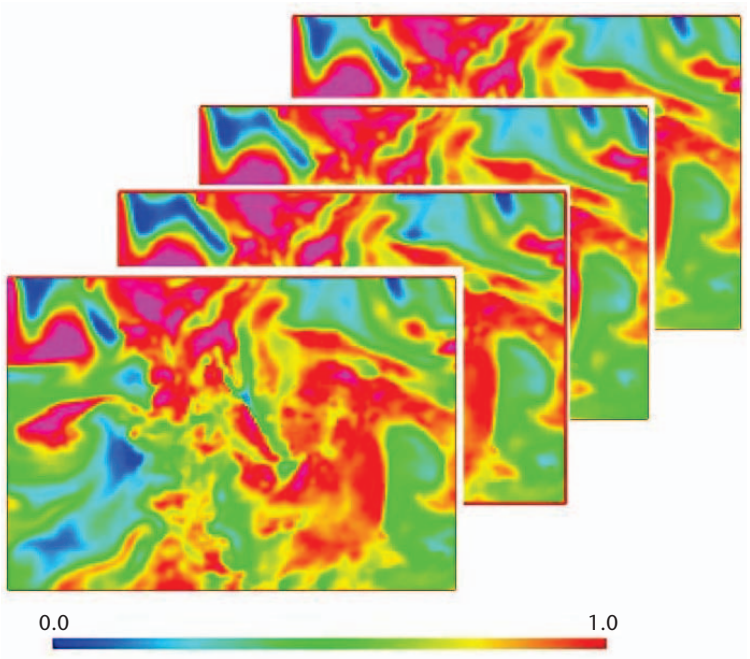
Figure 1a represents spatial domains of 0, 1, 2, and 3. For multivariate data, at each location in the spatial domain, there is a multivariate vector shown in Figure 1b. For multivalue data (denoted by square brackets in Figure 1c), at each location in the spatial domain, there can be univariate multivalue datum (left column in Figure 1c) all the way to multivariate multivalue datum (right column in Figure 1c).

The multiple values at each location and each point in time for each variable can be described by a probability density function (PDF). The PDF might be known or unknown, estimated from a sample, or approximated using a discrete function (a histogram) or a continuous

Published by the IEEE Computer Society 0272-1716/05/$20.00 © 2005 IEEE



1 Illustration and comparison between (a) multidimensional, (b) multivariate (at each location), and (c) multivalue (at each location) data.



2 Individual realizations of a 2D time varying multivariate weather forecast data set covering the continental US. Each time frame has multiple forecasts for each variable. Here, we show four different realizations or forecast scenarios of the humidity field for the first time frame. The values fall within [0, 1] and are colored using the standard rainbow color map.

function (a continuous PDF). Multivalue data can also be order invariant or order sensitive. For example, multivalue data derived from probabilistic models are not ordered. On the other hand, if the values represent a gene expression sequence, then a multivalue data set will only remain meaningful if the order of its components is preserved.

Motivation

Uncertainty visualization is the driving force behind our work. Accounting for uncertainty is important in several fields and has received significant attention within the geographic information system (GIS) community.[1] What is needed is a multifaceted characterization of the data and the process under which data is obtained. In practice, the characterizations are usually statistical in nature, typically resulting in scalar (for example, standard deviation) or multivariate (for example, additional statistical moments) descriptions of the uncertainty. Such representations usually end up as data attributes rather than being treated as an integral part of the data.

In cases where we have multiple measurements, experiments, or simulations, the collection can be taken together to simultaneously represent both data and its inherent uncertainty. As explained previously, we refer to the collection of values at each location as multivalue data. This data type represents both data and uncertainty at a location. Most of the previous work on visualizing uncertainty from the GIS community and ourselves[2] has not dealt with spatial multivalue data sets. Those that do take multivalues into account use animation to cycle through the realizations or features derived from the realizations, and have difficulty scaling up with the spatial resolution, spatial dimension, and even the number of values in a multivalue data set.[3,4]

Multivalue data arise in many disciplines. They occur in applications as diverse as geology, bioinformatics, engineering and manufacturing, oceanography, remote sensing, and ensemble weather forecasts to name a few. For a 0D or 1D multivalue data set, we can use a graph with a series of box plots for displaying them. However, as we go to two or higher dimensions, the current suite of visualization techniques is ill equipped to handle them. For example, scientists typically examine one instance of a multivalue data at a time (see Figure 2), thus failing to see the probabilistic or uncertain nature of their data. Another popular method is to collect and view statistical summaries of their multivalue data. However, this assumes that a few statistical summaries can properly describe their multivalue data, which might not be the case. Because there isn't a comprehensive set of visualization tools for looking at spatial multivalue data, particularly for two or more dimensions, this article looks at three possible strategies for displaying them with the third approach holding the most promise.

Data

We use different multivalue data sets to illustrate our proposed visualization techniques:

- 2D land cover data from a conditional simulation,
- 2D forest canopy data aggregated from multireturn lidar measurements,[5]
- 3D time varying multivariate ocean dynamics data where we look at one scalar field,
- another 3D multivariate ocean data set where we look at the multivalue baroclinic velocity field, and
- 2D time varying multivariate weather forecast data where we look at the multivalue velocity field.

The first data set is a synthetic example that was constructed using a small region in the Netherlands, imaged by the Landsat Thematic Mapper.[6] Assume the physical

variable of interest is the percentage of forest cover at each location, and that there are 150 ground truth points, as well as space-based measurements from the Landsat of a spectral vegetation index. This spectral vegetation index is related to the percentage of 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 1 pixel. The generated two 2D multivalue data sets using a conditional simulation algorithm. The first set, *sg2*, accounted for ground measurements only; while the second, *sg3*, used both ground measurements and the coincident satellite image. Both data sets consist of 101×101 pixels, where each pixel has 250 realizations. Each realization has values ranging from 0 to 255 rescaled from the percentage of forest cover.

The second data set is also 2D and pertains to forest canopy height. The forest canopy heights were measured using a multireturn lidar topographic imaging system carried on an aircraft flying over the Alexander Archipelago in Alaska. This equipment can retrieve up to five returns for every shot. The longitude and latitude of the elevation points were recorded with a GPS. For each 0.1 hectare cell, which is a 1,000-square-meter region, multiple returns were processed into 81 measures of forest canopy heights. The whole region scanned by the remote sensing system is divided into 69×47 cells. We refer to this as the *lidar* data.

The third data set is a 3D time varying output from ocean modeling. The model covers the Middle Atlantic Bight shelfbreak, which is about 100-km wide and extends from Cape Hatteras in North Carolina to the US–Canadian border. Both measurement data and ocean dynamics were 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, we employed an error subspace statistical estimation scheme.[7] This scheme is based on a reduction of the evolving error statistics to their dominant components or subspace. To account for nonlinearities, the method represents them with an ensemble of Monte Carlo forecasts. Hence, numerous 4D forecasts are generated and collected into a 5D field for each physical variable. For each physical variable, the dimensions of the data set are $65 \times 72 \times 42$ voxels, with multiple values at each point. We look at the sound speed field, which has 80 values at each voxel, and refer to this data set as *ocean*.

The fourth data set is a 3D multivalue multivariate data set. It covers a region from the Massachusetts Bay to the Cape Cod area off the US east coast. More than 200 physical and biogeochemical variables were measured. Forecasts and simulations were conducted for the period from 17 August to 5 October 1998. The area of study in the Massachusetts Bay was divided into a 53×90 grid; there are 600 values about each variable at each location in the grid. We examined the flow velocity in this data set, which we refer to as the *Massachusetts Bay* data set.

The fifth data set is an operational forecast from the National Oceanic and Atmospheric Association (NOAA). It is a 2D ensemble weather forecast available through http://wwwt.emc.ncep.noaa.gov/mmb/SREF/SREF.html. We refer to this data set as *sref*, which stands for short-range ensemble forecast. We used data generated on 24 October 2002. The ensemble is created from two different models, known as ETA and the regional spectral model (RSM), with five 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 coregistered and have different projections and spatial resolutions. Thus, for the purpose of this article, we use only the five-member ensemble from the RSM model. The RSM model's resolution is 185×129 and has 254 physical variables at each location, velocity being one of them. NOAA runs these forecasts twice a day, and for 22 different time steps during each run. We use this data set to illustrate flow visualization of multivalue velocity data.

Visualizing multivalue data

We propose three approaches to visualizing multivalue data sets. The first approach assumes that a statistical distribution can adequately represent the multivalue data. The second approach addresses the situation where that assumption does not hold and relies on shape descriptors to characterize each multivalue datum. The third approach provides a generalized methodology that uses mathematically and procedurally defined operators.

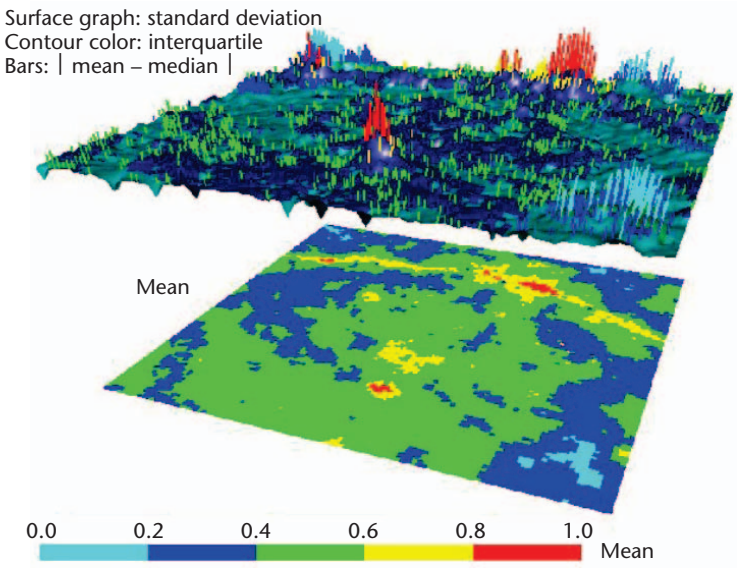
Parametric approach

The parametric approach to visualizing spatial multivalue data sets assumes that the distribution representing each multivalue datum can be adequately described by a few statistical parameters. That is, the distributions are assumed to have an underlying model such as Gaussian or Poisson. For instance, if we assume the underlying model is Gaussian, then the parametric approach calculates the mean, variance, and other statistical summaries that describe the data. For a single multivalue datum, Tukey's box plot glyph provides a compact representation that encodes minimum, maximum, mean, median, and quartile information. For spatial multivalue data, such as for 2D, a "2D box plot" is needed.

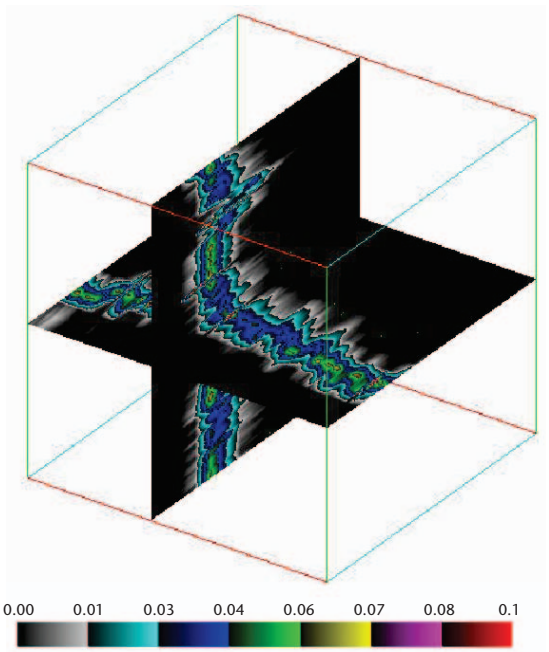
Literally rendering box plots over a spatial domain with transparent surfaces has some obvious problems. Instead, researchers in the GIS community, for example, display the statistical summaries using different themes or layers (see Figure 3, next page). This approach offers the advantage of familiarity with statistical parameters and ease of understanding. However, there are two serious drawbacks: Its basic assumption that parametric statistics can adequately represent the multivalue data might not be true; and it is difficult to extend the visualization beyond 2D spatial multivalue data sets. For example, this approach does not distinguish a unimodal distribution from a bimodal distribution if both have the same mean and variance. For 3D, you would have to perform separate volume renderings for each statistical parameter and correlate their locations and values, making for a difficult visualization task.

Shape descriptor approach

Parametric statistics don't always adequately repre-



3 Bottom plane is pseudocolored by the mean of the sg2 multivalue data. The bumps on the top surface are from the standard deviation and the surface is pseudocolored by the interquartile range. The bar glyphs show the differences between the mean and median of the multivalue data.



4 Two orthogonal slices from a histogram cube of the sg2 data set. The red axis is the spatial *X* dimension, the green axis is the spatial *Y* dimension, and the blue axis is the land cover value. Voxel colors are mapped to the density estimate for the land cover value at that location.

sent the distribution. For example, we can construct cases where two distributions have the same means and variances, but have drastically different shapes. This is generally the case for multimodal distributions where there might be multiple significant concentrations of values in the distribution. An alternative approach to visualizing spatial multivalue data sets then should strive to depict these aspects. One such method, presented in Kao et al.,[8] is to treat the data range as a third dimension and the density estimate of each multivalue datum as the voxel value. This creates a 3D volume that can be sliced and diced to reveal the locations and magnitudes of different peaks in each PDF (see Figure 4). In the figure, voxel colors are mapped to the density estimate for the land cover value at that location. The colors indicate value frequency. Black indicates zero frequency and red indicates high frequency. You can clearly see that most of the PDFs are unimodal with variations in the mean and variance, as evidenced by a single bluish swath across the spatial domain.

Peaks in a PDF indicate the most likely values a variable might take. We can describe peaks by a set of shape descriptors: number of peaks, and the height, width, and location of each peak.[9] Figure 5 illustrates how such shape descriptors might be displayed for a 2D slice of the ocean sound speed data set. While a bit cluttered, the image nevertheless tells us that most of the distributions are unimodal where the bottom plane is blue, and that their peaks are relatively higher, as indicated by the pinkish-reddish slab right above the blue regions. The few points that are multimodal also lie along the shelf break region indicated by the multicolored points on the bottom plane. Their corresponding peaks tend to be flatter and spread farther apart.

While this approach helps us see beyond the statistical parameters and into the shape of the distributions, the visualization techniques that both approaches afford cannot be readily extended to handle higher dimensional multivalue data. The next approach addresses that limitation.

Operator approach

This approach proposes a methodological treatment of multivalue data sets by defining operators for them. We propose procedural and arithmetic operators, as well as logical or similarity operators that work with multivalue data directly. These operators allow us to combine multivalue data together, for example, by adding or multiplying them. We can also define operators to promote a single value item to a multivalue datum, or to demote a multivalue datum to a single value. Likewise, we can define more complex operators to perform interpolation of multivalue data and compare multivalue data, gradient calculations, feature extractions, and other tasks. In short, they provide the necessary means for extending visualization techniques to handle multivalue data. The following discussion illustrates some of these operators and how we can use them to extend workhorse visualization techniques such as pseudocoloring, contour lines, isosurfaces, streamlines, and pathlines.

Pseudocolor. We introduce two basic classes of operators that are useful in pseudocoloring. In Equations 1 and 2, M denotes a multivalue data item. Scalar s and vector $\mathbf{v}$ are the results of these two operator classes:

$$s = ToScalar(M) \quad (1)$$

$$\mathbf{v} = ToVector(M) \tag{2}$$

We often use pseudocoloring to quickly distinguish different values in a scalar data set. It maps a range of values to a certain range of colors, usually in some linear fashion. A straightforward way to pseudocolor multivalue data is to convert multivalue data items into scalars. This can be achieved using the *ToScalar*(*M*) operator class. Different *ToScalar*() operators can be defined to suit the problem at hand. For example, it may be as simple as calculating the mean or variance.

Likewise, we can define *ToVector*() class operators in multiple ways. A simple example is illustrated in Figure 6 (next page) where we generated a 3-tuple from a multivalue data set. In the figure, the mean, which is bounded by [10, 108.3], is mapped to hue; standard deviation, ranging from [0, 48.6], is inversely mapped to value; and the absolute value of skewness, which is within [0, 4.1], is inversely mapped to saturation. Using this mapping, dark regions show higher uncertainty or standard deviation, while bright saturated points identify ground truth positions.

Contour lines. Contouring requires comparison of data values against a contour value. Since multivalue data sets, by their nature, have multiple values at each location, it does not make sense to compare their values against a single contour value. One possibility is to apply a *ToScalar*() operator to the multivalue data field and then run a traditional contouring algorithm. An alternative is to assume that the contour value is a multivalue data point. In this case, we need to determine whether two multivalue data points are the same.

For this purpose, we describe a class of similarity operators (see Equation 3) between two multivalue data points M_1 and M_2. When they are identical, the similarity operator returns $s = 0$. Like the two previous classes of operators, we can define similarity operators in multiple ways. Equations 4, 5, 6, and 7 show four different similarity operators. These are the absolute distance, Euclidean distance, Kolmogorov–Smirnov distance, and the Kullback–Leibler distance respectively. Equations 4 and 5 calculate the cumulative pairwise differences between two multivalue data points, which are being treated as distributions:

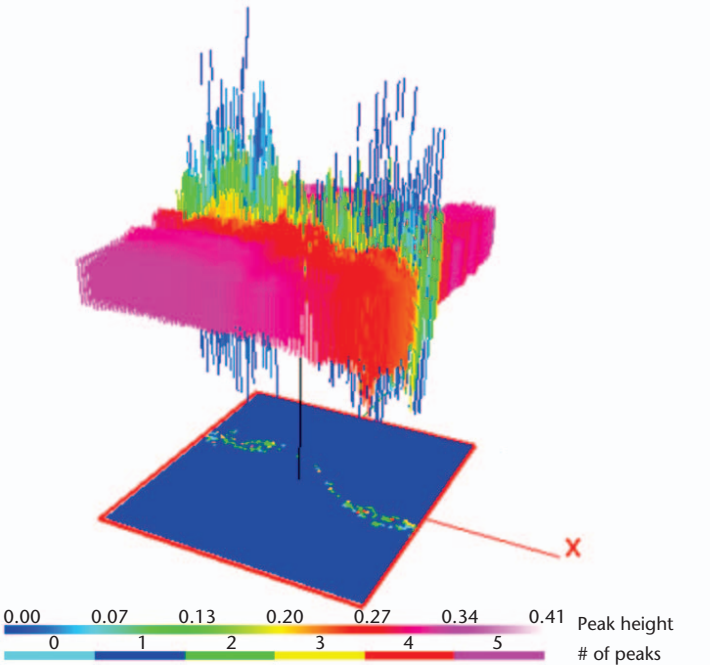
$$s = Similarity(M_1, M_2) \tag{3}$$

$$ABSD(M_1, M_2) = \int_{-\infty}^{+\infty} \left| M_1(x) - M_2(x) \right| dx \tag{4}$$

$$ED(M_1, M_2) = \left(\int_{-\infty}^{+\infty} \left(M_1(x) - M_2(x) \right)^2 dx \right)^{\frac{1}{2}} \tag{5}$$

$$KS(M_1, M_2) = \max \left| CDF(M_1(x)) - CDF(M_2(x)) \right| \tag{6}$$

$$KL(M_1, M_2) = \int_{-\infty}^{+\infty} M_1(x) \log \frac{M_1(x)}{M_2(x)} dx \tag{7}$$



5 The bottom plane is pseudocolored by the number of peaks in the ocean data. The lengths of the bar glyphs indicate the width of the peaks in the PDFs. The bar glyphs are pseudocolored by the heights of the peaks. Bright color indicates high peaks. The number of peaks varies between [0, 5], the width of the peaks varies between [0, 255], and the height of the peaks varies between [0, 0.41].

Because differences are taken pairwise, it would make sense that the two distributions cover the same range. Also note that two similarly shaped distributions that are offset from each other—such as the case with two normal distributions with the same variances but different means—would register as being quite dissimilar using these two distance measures.

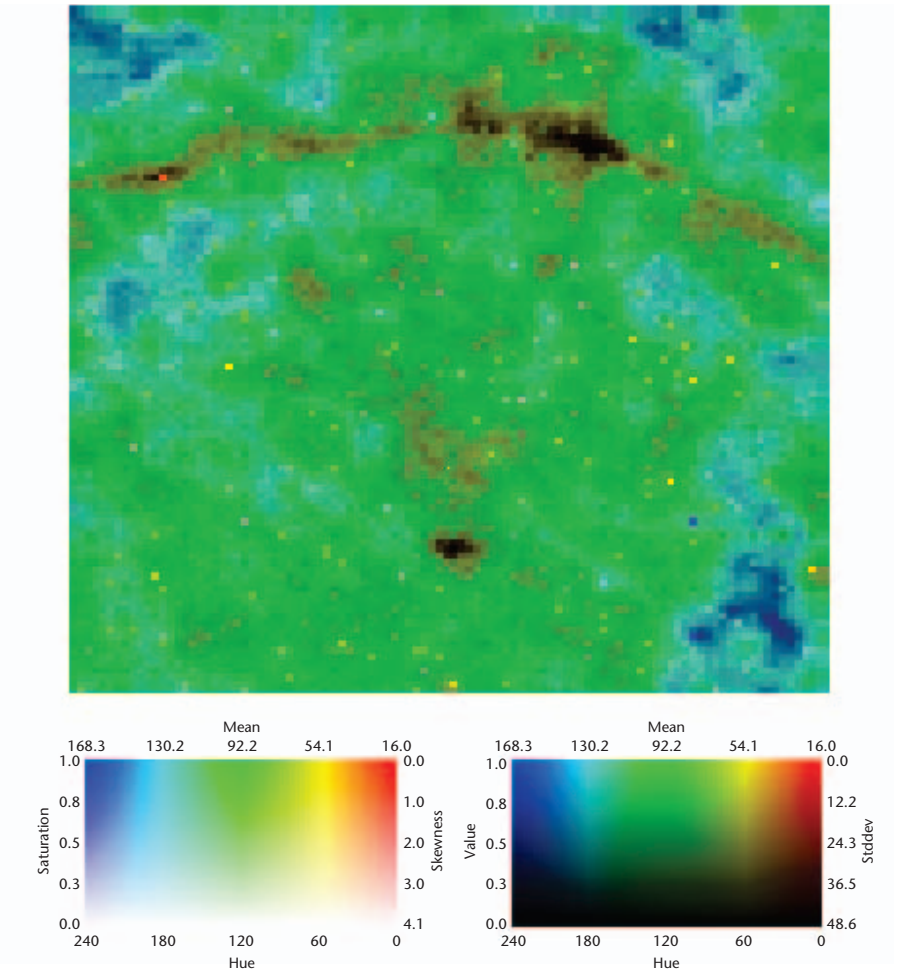
The *KS* distance measures the maximum distance between two cumulative distribution functions (CDF) and tests whether a data sample is consistent with a specified distribution function. When there are two data samples, the *KS* measures whether they come from the same distribution.[10]

The *KL* distance measures the relative entropy of M_1 with respect to M_2. Its use is quite popular in information theory and statistical mechanics. Although it is not a true metric, we can observe that the measure returns a non-negative value and it is 0 if both distributions are equivalent. Furthermore, the smaller the relative entropy, the more similar the two distributions.

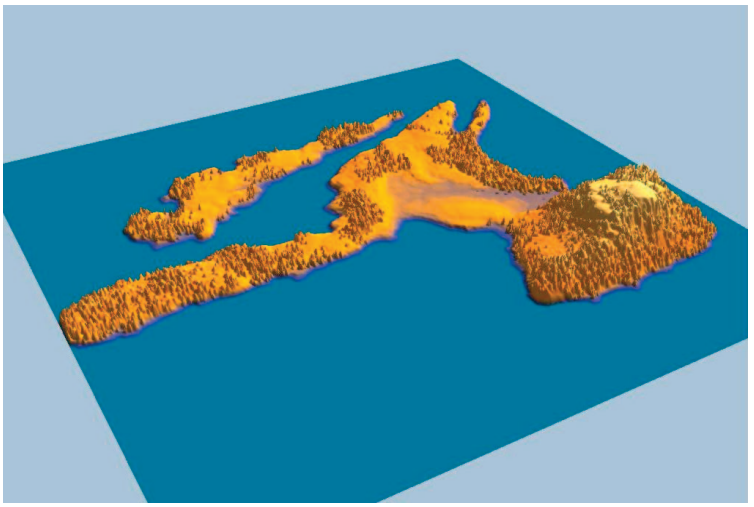
These similarity measures and other examples of operators are not exhaustive but rather illustrative of the broad possibilities from which they can be defined. We can look beyond statistics and information theory, for example, to signal processing, to select and define other similarity measures relevant to the application and task at hand.

In addition to comparing multivalue data points, we also need interpolation to find contour lines. For this, we introduce two more operators in Equations 8 and 9:

$$Scale(s, M) = (s \times x_1, s \times x_2, s \times x_3, \ldots, s \times x_n) \tag{8}$$



6 **An HSV pseudocolor rendering of the multivalue data sg2 obtained using a *ToVector(M)* operation that extracts the mean, standard deviation, and skewness of each multivalue datum.**



Courtesy of Marc Kramer

7 **Graphical model of the Alaska High Island forest from the lidar data. Individual trees in the forest are represented by the tree-like icons.**

$$Interp(M_1, M_2, s) = Scale(1-s, M_1) \oplus Scale(s, M_2) \quad (9)$$

The $\oplus$ operation represents an addition of two multivalue data items. Again, there is more than one way to define such an operation. To illustrate, we consider two alternative definitions. The first is due to Gerasimov et al., which we refer to as *convolution addition* defined in Equation 10.[11] The second one is from Gupta and Santini, which we refer to as *binwise addition* defined in Equation 11.[12]

Convolution addition is statistically meaningful when PDFs describe the two multivalue data items to be added. Let P be the PDF of random variable x, and Q be the PDF of random variable y. The addition of these two independent PDFs results in another PDF for the sum of both random variables. This PDF is defined in Equation 10, where $z = x + y$ and R is the convolution sum of P and Q:

$$R(z) = \int_{-\infty}^{+\infty} P(x)Q(z-x)dx \quad (10)$$

Binwise addition does not require the multivalue data items to be PDFs, but does require that they both be evaluated over the same range of values and have the same number of bins—such as in histograms representing different PDFs. Let M_1 and M_2 denote two multivalue data items about variable x. Their corresponding bins are then added up to form a new histogram where each bin contains the sum. This was introduced by Gupta and Santini[12] as

$$R(x) = M_1(x) + M_2(x) \quad (11)$$

Given a multivalue contour target, which we shall denote as T, multivalue contouring requires finding a set of multivalue data points in the field that matches the target. In a discretized grid, this means finding intersections of cell edges with the target. Intersections can be found by setting a threshold t on the similarity measure between the target and interpolated multivalue data points on an edge. At each intersection, an interpolated multivalue data lies within a distance t of the target. This relaxes the definition so that all the multivalue data points on a contour line are within a distance t of the target.

One challenge when finding intersections is that some similarity measures, for example KL, are not linear metrics. We therefore provide the following modifications. Let two adjacent multivalue data points in the field be denoted by M_1 and M_2. Let $Sim(M_1, M_2)$ denote a similarity operator used to compute distances between multivalue data items. Let the distance from M_1 and M_2 to T be $a = Sim(M_1, T)$ and $b = Sim(M_2, T)$, respectively. If the threshold t is in the range $[a, b]$, then a multivalue data item at a distance t from the multivalue contour item lies somewhere between M_1 and M_2. We then subdivide the edge between M_1 and M_2 into n intervals, and gen-

erate interpolated multivalues $I_1, \ldots, I_n$, using Equation 9. Each of these multivalues are then compared to T. The edge intersection is set at the location of the multivalue data with the smallest distance from T. We describe the details of this procedure in the following algorithm for contour line interpretation:

```
for all adjacent pairs of multivalue data items M1
  and M2 do
  if t ∈ [Sim(M1, T), Sim(M2, T)] then
     subdivide along M1 and M2 to obtain n
      interpolated multivalue data items I1, I2 ..., In
     dmin ← +∞
     for i ← 1..n do
        d ← Sim(Ii, T)
        if d < dmin then
           dmin ← d
           s ← i
        end if
     end for
     s is the interpolation point between
     M1 and M2
  end if
end for
```

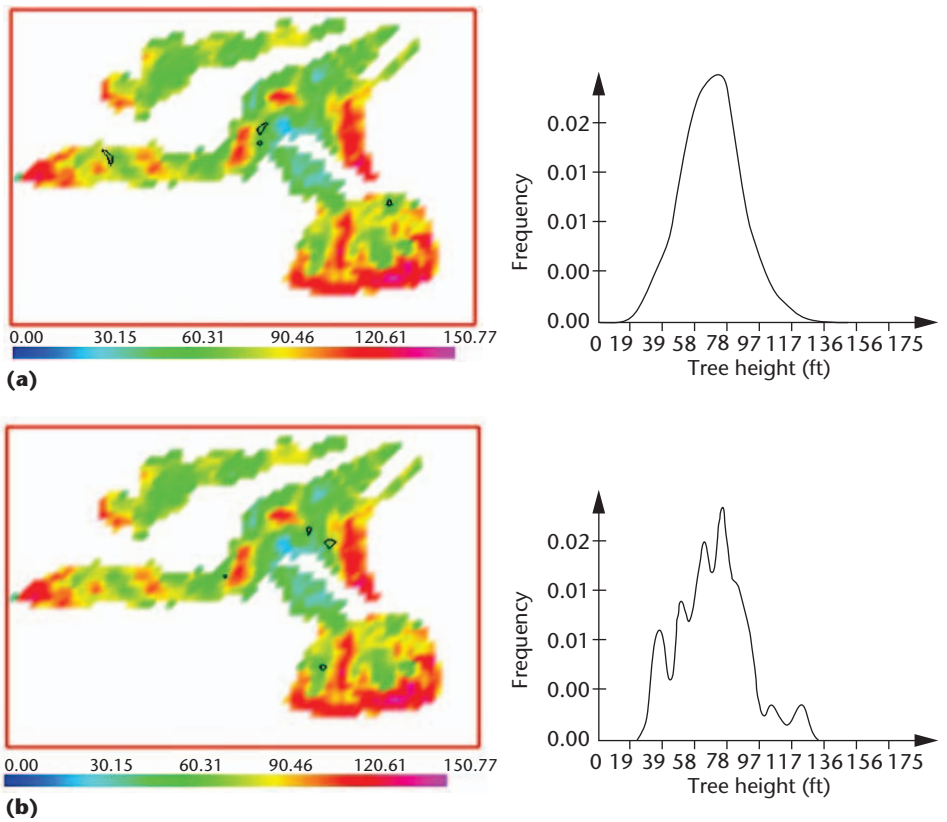
The lidar data set was collected over Alaska High Island and the topography of this region is illustrated in Figure 7. Figure 8 shows contouring results on the lidar data with contour lines generated to match characteristics of the target shown on the right. We used the absolute value of difference (*ABSD*) similarity operator here. White areas represent ocean. The land is colored using the mean of distributions. Blue regions indicate distributions with low mean. This figure also illustrates that pseudocoloring based on the mean alone is not sufficient to identify regions with similar distribution patterns as the target.

Because we are dealing with multivalue data sets, contour lines can mean different things. In one instance, when the multivalue data are first demoted to single values, the contour lines take on the traditional meaning of lines equal to the contour value. In the second instance (see Figures 8a and 8b), when the target itself is a multivalue, then the contour lines are places where the multivalue data points are similar to the target. We can devise further variations to depict additional properties of multivalue data sets as the need arises. For example, if we were to look for multivalue data sets whose PDF looks similar to the target in Figure 8a but with a different mean tree height, then a score could be generated for shifts in the mean within a certain range. This is illustrated in the following algorithm for multivalue matching with shifting:

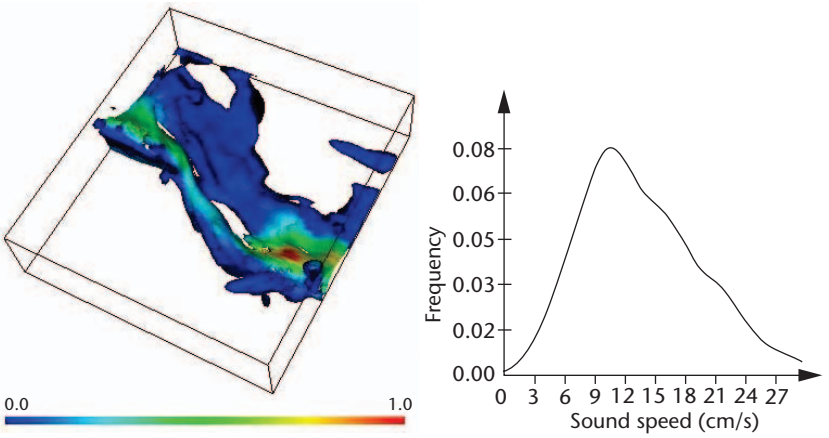
```
dmin ← +∞
for all distributions Mi in the domain do
  for j ← –S..S do
     shifted T ← vary the mean of T by j intervals
     d ← Sim(Mi, shiftedT)
     if d < dmin then
        dmin ← d
     end if
  end for
  di ← dmin
end for
```

where d_i, d, and d_{min} are distances computed using the similarity operator $Sim(M_i, M_2)$, and $\pm S$ is the range of mean offsets for M_i.

Isosurfaces. Isosurfaces are the 3D counterpart for contour lines. In Figure 9 (next page), we illustrate an isosurface generated by comparing each of the multivalue data in the volume against the multivalue target shown on the right. The data are the multivalue sound speed from the ocean data set. We use the *KL* similarity operator (see Equation 7) instead of the *ABSD* operator. Because searching for the edge intersection is an expensive process that is proportional to how finely each edge is subdivided, and the cost for the 3D case is a degree



8 Multireturn lidar data showing tree heights: (a) Unimodal target with threshold set to 0.11. Contours correspond to trees with similar ages (tree heights centered around 78 feet). (b) Multimodal target with threshold set to 0.14. Contours represent trees with similar age mixtures.



9 An isosurface produced in the ocean data set using the target distribution (shown on the right) chosen from the shelfbreak region. The surface represents regions in the data where the multivalue data items are similar to the target distribution. The *KL* operator was applied and the threshold was 0.15. Color can also be used to display certain properties of the multivalue data items by using an appropriate *ToScalar*(*M*) operator. Here, we color the surface by the standard deviation of the multivalue data at each voxel.

more than the 2D case, the isosurface shown in Figure 9 is actually an approximation. Unlike the case for 2D contours, we use a standard marching cubes algorithm on the scalar volume returned by comparing each multivalue data with the target. That is, for the sake of reducing computational cost, we assume that the *KL* operator is a linear metric, at least for the case when the two multivalue data points along an edge are close to each other. Of course, we could use the same subdivision method as the one used in the 2D case.

Streamlines. Streamline generation is a standard flow visualization technique. Streamlines are generated by integrating the path of massless particles as they are carried through a velocity field. For illustration purposes, we look at how the simple Euler integration, shown in Equation 12 , can generate streamlines in multivalue velocity fields:

$$P_{i+1} = P_i + \mathbf{v}(P_i)\Delta t \tag{12}$$

where P_i is the current position at time step i, $\mathbf{v}(P_i)$ is the velocity at P_i, and Δt is the integration time step.

Given a seed point P_0, the velocity at that location is a multivalue vector. That is, there are multiple possible trajectories resulting in different streamline paths. In the succeeding time steps, each trajectory will have their own set of multivalue velocities to deal with—that is, P_i and P_{i+1} are multivalue data points. The possibilities appear to grow exponentially. These can all be brought under control using different interpretations of the equations. At minimum, we will need to define that a multiplication between a scalar s and a multivalue datum M results in a multivalue datum M' where each member of the multivalue datum is multiplied by the scalar quantity $M' = sM$.

A simple interpretation of the Euler equation is to apply a *ToScalar*() operator to each of the multivalue data quantity at each time step. If the *ToScalar*() operator is to take the mean of the multivalue datum, then we end up with a streamline that would represent the particle's mean path. That is, the mean velocity at P_i is multiplied by Δt and added to the seed point to obtain a single valued position P_{i+1}, and the process is repeated.

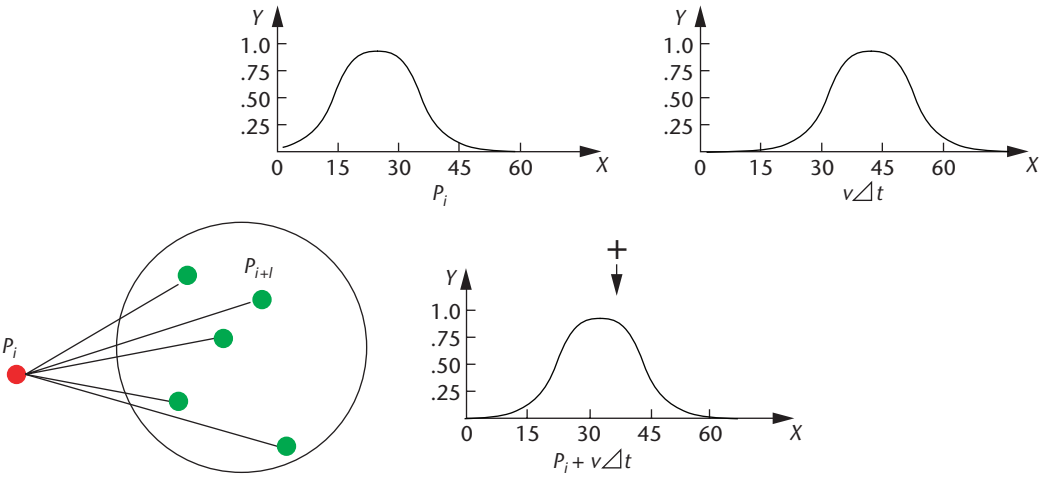
A slightly different interpretation is to apply the *ToScalar*() operator only after the completion of each integration step. That is, starting from the seed point, we follow the different trajectories for one step resulting in a P_{i+1} term that is a multivalue position. A *ToScalar*() operator is then applied to this to get the centroid of these points before the next iteration proceeds, then the process is repeated.

Yet another interpretation is to promote each of the components (except Δt) to a multivalue data at each integration step. In this case, the seed point is first promoted to a multivalue location by simply replicating its values n times. This then requires an addition ($\oplus$) of two multivalue data, which results in P_{i+1} being a multivalue location. The process is then repeated in subsequent iterations.

Figure 10 illustrates the process of generating streamlines using convolution addition. At each integration step, P_i and $v \times \Delta t$ (represented by their respective PDFs) are added together to obtain another distribution representing P_{i+1}. For convolution addition, corresponding samples drawn from $v \times \Delta t$ are added up according to their distributions. During rendering, a white transparent circle is drawn around the resulting sum. This is illustrated in Figure 10 by the circle surrounding the samples of P_{i+1}. This procedure is repeated until the center of the streamline circle goes out of bounds, or a predefined number of integration steps has been reached.

Because multivalue velocity fields naturally capture the uncertainty or variability in the flow, streamlines do not and should not be single well-defined paths that might falsely imply a sense of certainty. Rather, streamlines in multivalue velocity fields should show the general trajectories. To achieve the desired effect, we draw overlapping transparent circles after each integration step. The circles are sized and positioned so that they circumscribe the set of possible positions for each integration step. An additional parameter S can also be added to allow the user to vary the size of these circles by specifying to keep only S percent of the points closest to the center of each circle.

Figure 11a shows the current practice of drawing spaghetti plots where streamlines are generated independently using each velocity field from the multivalue vector field and overlaying them one on top of the other. The black streamline is generated by averaging the positions of the particles at each iteration



10 Illustration of the convolution addition in generating streamlines.

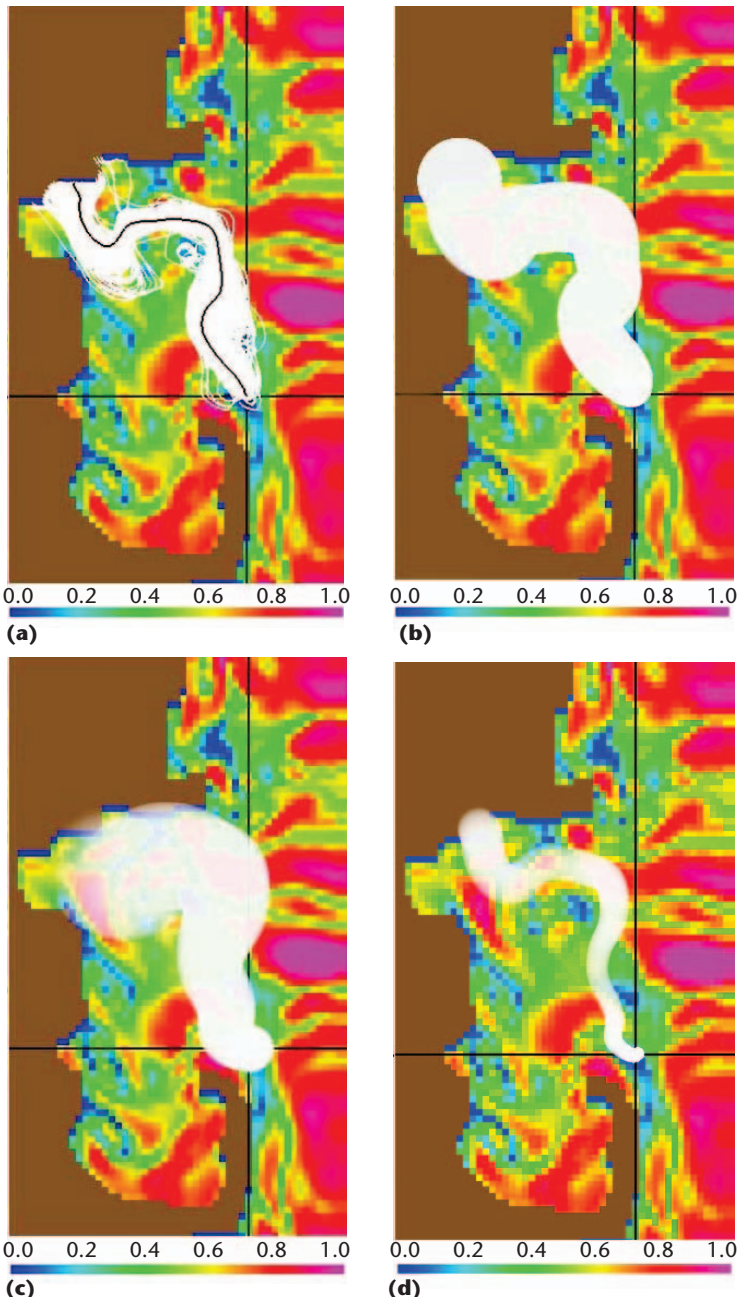
and represents the mean streamline from the seed point. The background is colored by the likelihood of finding critical points. Likelihood values range from 0 to 1, and are colored using the standard rainbow map. Blue regions indicate places where critical points are likely to be found. The brown region indicates land. Figures 11b to 11d illustrate the appearance of multivalue streamlines rendered using transparent white circles as described previously. They convey the variability or uncertainty of the streamline as defined by the multivalue vector field.

Pathlines. Pathlines trace the path of a particle over a time varying flow field.[13] Again, using Euler integration for the sake of simplicity in illustrating how time varying multivalue velocity fields can be visualized, the relevant integration equation is

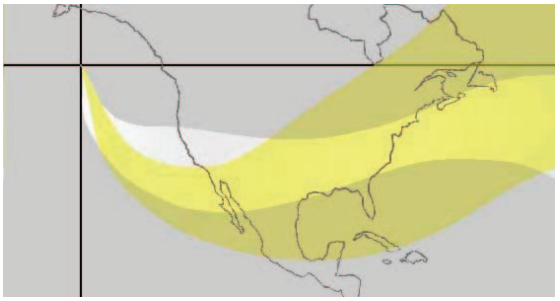
$$P_{i+1} = P_i + \mathbf{v}(P_i, t)\Delta t$$

That is, the velocities at the current time frame t, as opposed to a static velocity field, are used to determine the position(s) of the particle(s) in the next time frame.

Figure 12 illustrates a pathline from an ensemble of time varying flow data . Pathlines can be rendered with semitransparent circles in the same way streamlines are rendered. The sref ensemble forecast data has 22 time steps for each forecast run. However, each time step is three hours apart, which is too large for numerical integration. So, we



11 Streamlines from the Massachusetts Bay data set generated from the same seed point. (a) Conventional spaghetti plot. The black curve indicates the mean of the 600 individual streamlines. (b) Streamline circles generated using binwise addition. (c) Streamline circles generated using convolution addition. (d) Streamline circles generated using convolution addition with 30 percent of the radius in (c).



12 White trace represents a streamline trajectory of the first time frame from the sref data set. The yellow trace represents a pathline trajectory using the same seed point as the streamline. Both traces are the results of binwise addition.

conducted temporal interpolation of the ensemble flow field. For the example in Figure 12 , we introduced 30 additional time frames within each 3-hour forecast period using the interpolation strategy for multivalue data items described in Equation 9. This procedure produced an ensemble velocity field every 0.1 hour, allowing us to use the same integration step in our simple fixed step Euler integration. We also used binwise addition. Figure 12 shows a streamline and a pathline generated from the same seed point and initial time frame. The streamline is rendered in white, while the pathline is rendered in yellow. The two trajectories are significantly different from each other. The pathline trajectory is much wider than the streamline trajectory, which indicates the higher temporal and spatial variability compared to a steady flow.

Discussion

We listed the three approaches for visualizing spatial multivalued data sets in the order of their development as well as generality. While the operator based approach is the most flexible and powerful approach for handling multivalue data, it also requires additional training in its use and interpretation.

So, how do you choose among the multitude of alternatives? Obviously, there are a number of factors that come into play including the application domain of where the data came from, the task at hand, and whether a particular operator made sense. Even with these filters to reduce the number of choices, you still have several reasonable operators to choose from. At this point, experience comes into play.

Absent that, systematic experimentation and evaluation are needed to compare the alternatives. For example, you can first assume that the outcome produced by an operator, corroborated by most of the alternative operators, is more likely to be correct than the outcome with little corroboration among operators. This process can further reduce the candidate set of operators. The remaining candidates can then be ranked by how well they perform on a known data set, or how consistently they perform, or how efficient it is to compute, and so on. While the operator-based approach is powerful and flexible, it does require some sophistication and care from the user.

There is other interesting work that needs to be done using the operator-based approach. Some of the extensions we are looking at include volume rendering multivalue data sets, interpolation of spatially sparse multivalue data, and feature extraction of multivalue data sets. The operator-based approach is particularly important when dealing with large time varying multidimensional, multivariate, multivalue data sets as they are a factor *n* times larger than their single value counterpart. ■

Acknowledgments

This work was supported in part by the NASA Intelligent Systems Program under Cooperative Agreement NCC2-1260 and NSF ACI-9908881. We 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. Finally, we thank the reviewers for their helpful comments.

References

1. M.K. Beard, B.P. Buttenfield, and S.B. Clapham, *NCGIA Research Initiative 7: Visualization of Spatial Data Quality*, tech. paper 91-26, Nat'l Center for Geographic Information and Analysis, Oct. 1991, pp. 59; http://www.ncgia.ucsb.edu/Publications/Tech_Reports/91/91-26.pdf.
2. A. Pang, C.M. Wittenbrink, and S.K. Lodha, "Approaches to Uncertainty Visualization," *The Visual Computer*, vol. 13, no. 8, 1997, pp. 370-390; http://www.cse.ucsc.edu/research/avis/unvis.html.
3. C.R. Ehlschlaeger, A.M. Shortridge, and M.F. Goodchild, "Visualizing Spatial Data Uncertainty Using Animation," *Computers & Geosciences*, vol. 23, no. 4, 1997, pp. 387-395.
4. R.M. Srivastava, *The Visualization of Spatial Uncertainty, Stochastic Modeling and Geostatistics: Principles, Methods, and Case Studies*, J.M Yarus and R.L. Chambers, eds., American Assoc. of Petroleum Geologists, 1994.
5. G.J. Nowacki and M.G. Kramer, "The Effects of Wind Disturbance on Temperate Rain Forest Structure and Dynamics of Southeast Alaska," *General Technical Report PNW-GTR-421*, US Dept. of Agriculture, Forest Service, Pacific Northwest Research Station, April 1998.
6. J.L. Dungan, "Conditional Simulation: An Alternative to Estimation for Achieving Mapping Objectives," *Spatial Statistics for Remote Sensing*, Kluwer, 1999, pp. 135-52.
7. 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*, vol. 127, no. 7, 1999, pp. 1408-1432.
8. D. Kao, J. Dungan, and A. Pang, "Visualizing 2D Probability Distributions from EOS Satellite Image-Derived Data Sets: A Case Study," *Proc. Visualization 2001*, IEEE CS Press, 2001, pp. 457-460.
9. D. Kao et al., "Visualizing Spatially Varying Distribution Data," *Proc. 6th Int'l Conf. Information Visualization*, IEEE CS Press, 2002, pp. 219-225.
10. W.T. Eadie, *Statistical Methods in Experimental Physics,* North-Holland Pub. Co., 1971.

11. V.A. Gerasimov, B.S. Dobronets, and M. Yu. Shustrov, "Numerical Operations of Histogram Arithmetic and Their Applications," *Automation and Remote Control*, vol. 52, no. 2, 1991, pp. 208-212.
12. A. Gupta and S. Santini, "Toward Feature Algebras in Visual Databases: The Case for a Histogram Algebra," *Advances in Visual Information Management: Visual Database Systems*, Kluwer, 2000, pp. 177-196.
13. D.L. Darmofal and R. Haimes, "An Analysis of 3D Particle Path Integration Algorithms," *J. Computational Physics*, vol. 123, no. 1, 1996, pp. 182-195.

Alison L. Love *is a software engineer at Adobe Systems Incorporated. Her research interests include scientific visualization, image processing, and computer animation. Love received a BS in computer science and BA in economics from Peking University and a PhD in computer science from the University of California, Santa Cruz. Contact her at alove@adobe.com.*

Alex Pang *is a professor in computer science at the University of California, Santa Cruz. His research interests are in comparative and uncertainty visualization, flow and tensor visualization, and collaborative visualization. Pang received a BS in industrial engineering from the University of the Philippines and a PhD in computer science from UCLA. Contact him at pang@cse.ucsc.edu.*

David L. Kao *is the assistant branch chief and a research scientist in the NASA Advanced Supercomputing Division at NASA Ames Research Center. His research interests include scientific visualization, information visualization, numerical flow visualization, and computer graphics. Kao received a PhD in computer science from Arizona State University. He is an associate editor of* IEEE Transactions on Visualization and Computer Graphics. *Contact him at David.L.Kao@nasa.gov.*