Visualizing Spatial Multi-value Data

Alison Love¹, David L. Kao², and Alex Pang¹
¹Computer Science Department, UCSC
²NASA Ames Research Center
{alison,pang}@soe.ucsc.edu, David.L.Kao@nasa.gov

Abstract

Ensemble forecasts, outcomes from conditional simulations, or repeated measurements in an experiment all produce multiple instances of the same physical field. We refer to this as a multi-value data type. Specifically, a multi-value data point contains multiple values for a single variable. If there is a single multi-value data point, then visualizing it can be carried out using a simple graph either showing the different values, or the frequency of the different values. However, if the multi-value data exist over a spatial domain, then the existing suite of visualization techniques has limited power in visualizing them. This paper introduces the multi-value data type, and suggests three different ways of visualizing spatial multi-value data sets.

1 Introduction

We introduce multi-value data as a new data type in the context of scientific visualization. While this data type has been in existence in other fields, it has largely been ignored by the visualization community. Formally, a multi-value 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 may arise from a measurement process or a modeled process. In the latter case, it is particularly useful to consider probabilistic models where the collection of values describe the set of possible outcomes of the modeled process.

Multi-value data sets can be defined for multiple dimensions. A spatial multi-value data set consists of a multi-value datum at each physical location in the domain. The time dimension is equally valid. This leads to spatio-temporal multi-value data sets where one has a time varying, multidimensional data with a multi-value datum at each location and time.

To illustrate the challenge of visualizing spatial multi-value data sets, consider having to visualize a time varying volumetric temperature field. One 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 multi-value data set would be very difficult to comprehend by sequentially watching these animations.

Compounding matters, the previous example was just a simple scalar multi-value data set. We also need to consider vector, or more generally, multivariate multi-value data sets. Specifically, another data descriptor is whether a data set is univariate or multivariate. A multivariate multi-value data set has a multi-value 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 many different disciplines, we want to make a clear distinction between their usage in this paper. 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 visualization.

Note that while both multi-value 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 may be applied to multi-value 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 latter is how a vector representing a multi-value data type would be interpreted by a multivariate visualization task.

To recapitulate, we have three orthogonal data descriptors: multi-value, multivariate, and multidimensional. A complex data set can have all three properties leading to a multidimensional, multivariate, multi-value data set. These concepts are illustrated in Figure 1.

![Figure 1: Illustration and comparison between multidimensional, multivariate and multi-value data.](image)

The multiple values at each location and/or each point in time and/or each variable can be described by a probability density function (pdf). The pdf may be known or unknown, estimated from a sample, or approximated using a discrete function (a histogram) or a continuous function (a continuous pdf). Multi-value data can also
be order-invariant or order-sensitive. For example, multi-value data from probabilistic models are not ordered. On the other hand, if the values represent a gene expression sequence, then a multi-value data set will only remain meaningful if the order of its components is preserved.

## 2 Motivation

Uncertainty visualization is the driving force behind this work. Accounting for uncertainty is of prime importance within the GIS (Geographic Information System) community [1]. It is a multi-faceted characterization of the data and process under which data is obtained. Its descriptors are usually statistical in nature, typically resulting in scalar (e.g. standard deviation) or multivariate (e.g. 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 one has multiple measurements, experiments, or simulations, the collection can be taken together to simultaneously represent both data and its inherent uncertainty. As explained in section 1, we refer to the collection of values at each location as a multi-value data. It represents both data and uncertainty at a location. Most of the previous work on visualizing uncertainty from the GIS community and ourselves [15] has not dealt with spatial multi-value data sets. Those that do, for example [6, 16], 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 multi-value data set. This paper describes three approaches for looking at multi-value data sets, with the third approach holding the most promise.

Multi-value 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 zero dimensional or one dimensional multi-value data set, one can use a graph with a series of box plots to display them. However, as one goes to two or higher dimensions, the current suite of visualization techniques are ill equipped to handle them. For example, scientists typically examine one instance of a multi-value 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 multi-value data. However, this assumes that their multi-value data can be properly described by a few statistical summaries, which may not necessarily be the case. Because there isn’t a comprehensive set of visualization tools for looking at spatial multi-value data, particularly for two or more dimensions, this paper looks at three possible strategies for displaying them. But first, we briefly describe the data sets used in the subsequent discussions.

## 3 Data

Different multi-value data sets are used to illustrate the proposed visualization techniques. These are:

- 2D land cover data from conditional simulation;
- 2D forest canopy data aggregated from multi-return lidar measurements [14];
- 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 multi-value baroclinic velocity field;
- and 2D time varying multivariate weather forecast data where we look at the multi-value velocity field.

The first data set is a synthetic example constructed using a small region in the Netherlands, imaged by the Landsat Thematic Mapper [4]. 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 one pixel. Two, 2D multi-value data sets were generated using a conditional simulation algorithm [3]. The first, \(sg2\), accounted for ground measurements only; while the second, \(sg3\), used both ground measurements and the coincident satellite image. Both data sets consist of 101 \(\times\) 101 pixels, where each pixel has 250 realizations. Each realization has values ranging from 0 to 255 rescaled from percentage forest cover.

The second data set is also 2D and pertains to forest canopy height. Forest canopy heights were measured using a multi-return lidar topographic imaging system, carried on an aircraft flying over the Alexander Archipelago in Alaska. The equipment can retrieve up to five returns for every shot it makes. The longitude and latitude of the elevation points are recorded with a Global Positioning System (GPS). For each 0.1 hectare cell, which is a 1,000 m² region, multiple returns are processed into 81 measures of forest canopy heights. The whole region scanned by the remote sensing system is divided into 69 \(\times\) 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 to Canada. 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, an Error Subspace Statistical Estimation (ESSE) scheme [12] was 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.

Figure 2: Individual realizations of a 2D time varying multivariate weather forecast data set covering the continental US. For each time frame, there are multiple forecasts for each variable. Here, we show four different realizations of forecast scenarios for the humidity field. The values fall within [0,1] and are colored using the standard rainbow colormap.
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 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 multi-value multivariate data set. It covers a region from the Massachusetts Bay to the Cape Cod area off the U.S. East coast. Over 200 physical and bio-geochemical variables are measured. Forecasts and simulations were conducted for the period from August 17 -- October 5, 1998. The area of study in the Massachusetts Bay was divided into a $53 \times 90$ grid, and 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 NOAA. It is a 2D ensemble weather forecast available through http://www.emc.ncep.noaa.gov/mmb/SREF/SREF.html. The data set is referred to as sref, which stands for short-range ensemble forecast. We used data that was generated on October 24, 2002. The ensemble is created from two different models, known as ETA and 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 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 $185 \times 129$ and has 254 physical variables at each location, velocity being one of them. The forecast is run twice a day, and for 22 different time steps during each run. We use this data set to illustrate flow visualization of multi-value velocity data.

![Figure 3: The bottom plane is pseudo-colored by the mean of the sref multi-value data. The top surface is drawn using the standard deviation and it is pseudo-colored by the interquartile range. The glyph bars show the differences between the mean and median of the multi-value data.](image)

4 Visualizing Multi-value Data

We propose three approaches to visualizing multi-value data sets. The first approach assumes that the multi-value data can be adequately represented by a statistical distribution and can therefore be described by a few parametric statistics. The second approach addresses the situation where that assumption does not hold, and relies on shape descriptors to characterize each multi-value datum. Finally, the third approach provides a generalized methodology that uses mathematically and procedurally defined operators.

4.1 Parametric Approach

The parametric approach to visualizing spatial multi-value data sets assumes that the distribution representing each multi-value 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 multi-value datum, Tukey’s box plot glyph [13] provides a compact representation that encodes minimum, maximum, mean, median, and quartile information. For spatial multi-value 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, what has been done, for example in the GIS community, is to display the statistical summaries using different themes or layers. This is also illustrated in Figure 3. This approach offers the advantage of familiarity with statistical parameters and ease of understanding. However, there are two serious drawbacks: its basic assumption that the multi-value data can be adequately represented by parametric statistics may not be true; and difficulty in extending the visualization beyond 2D spatial multi-value 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, one would have to perform separate volume renderings for each statistical parameter, and correlate their locations and values, making for a very difficult visualization task.

4.2 Shape Descriptor Approach

Parametric statistics do not always adequately represent the distribution. For example, one 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 may be multiple significant concentrations of values in the distribution. An alternative approach to visualizing spatial multi-value data sets then should strive to depict these aspects. One such method, presented in Kao et al. [9], is to treat the density estimate of each multi-value datum as the voxel value and the data range of the density estimate as a third dimension. 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).

Peaks in a pdf indicate the most likely values a variable might take. Peaks can be described by a set of shape descriptors: number of peaks, and the height, width and location of each peak [10]. Figure 5 illustrates how such shape descriptors might be displayed for a 2D slice of the ocean sound speed data set.

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 multi-value data. The next approach addresses that limitation.

4.3 Operator Approach

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

**Pseudo-color**

We introduce two basic classes of operators that are useful in pseudo-coloring. In Equations 1 and 2, $M$ denotes a multi-value data item. Scalar $s$ and vector $v$ are the results of these two operator classes.

\[
\begin{align*}
    s &= \text{ToScalar}(M) \\
    v &= \text{ToVector}(M)
\end{align*}
\]

Pseudo-coloring is often used 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 pseudo-color multi-value data is to convert multi-value data items into scalars. This can be achieved using the $\text{ToScalar}(M)$ operator class. Different $\text{ToScalar}()$ operators can be defined to suit the problem at hand. For example, it may be as simple as calculating the mean or variance. Likewise, $\text{ToVector}()$ class operators can be defined in multiple ways. A simple example is illustrated in Figure 6 where a 3-tuple is generated from a multi-value data set.

**Contour Lines**

Contouring requires comparison of data values against a contour value. Since multi-value 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 $\text{ToScalar}()$ operator to the multi-value data field and then run a traditional contouring algorithm. An alternative is to assume that the contour value is a multi-value data point. In this case, we need to be able to determine if two multi-value data points are the same or not. For this purpose, we describe a class of similarity operators (Equation 3) between two multi-value data points $M_1$ and $M_2$. When they are identical, the similarity operator returns $s = 0$. Like the two previous classes of operators, there are multiple ways of defining similarity operators. 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 \[11\] respectively. Equations 4 and 5 calculate the cumulative pairwise differences between two multi-value data points, which are being treated as distributions. 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 is used to test whether or not a data sample is consistent with a specified distribution function. When there are two samples of data, it is used to test whether or not these two samples come from the...
The $KL$ operator represents an addition of two multi-value 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. [7] which we refer to as “convolution addition” defined in Equation 10. The second one is from Gupta and Santini [8] which we refer to as “binwise addition” defined in Equation 11.

1. Convolution addition: This addition is statistically meaningful when the two multi-value data items to be added are described by pdfs. 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. It is defined in Equation 10, where $z = x + y$ and $R$ is the convolution sum of $P$ and $Q$.

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

2. Binwise addition: This addition does not require the multi-value 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 multi-value 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 in [8] as:

$$ R(x) = M_1(x) + M_2(x) $$

Given a multi-value contour, which we shall call the target, multi-value contouring requires finding a set of multi-value 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 multi-value data points on an edge. At each intersection, an interpolated multi-value data lies within a distance $t$ of the target. This relaxes the definition so that all the multi-value 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 $T$ denote the multi-value target, and let two adjacent multi-value data points in the field be denoted by $M_1$ and $M_2$. Let $\text{Sim}(M_1, M_2)$ denote a similarity operator used to compute distances between multi-value data items. Let the distance from $M_1$ and $M_2$ to $T$ be $a = \text{Sim}(M_1, T)$ and $b = \text{Sim}(M_2, T)$ respectively. If the threshold $t$ is in the range $[a, b]$, then a multi-value data item at a distance $t$ from the multi-value 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 generate interpolated multi-values $I_1, \ldots, I_n$, using Equation 9. Each of these multi-values are then compared to $T$. The edge intersection is set at the location of the multi-value data with the smallest distance from $T$. The details of this procedure are described in Algorithm 1.
(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.

Figure 8: Multi-return lidar data showing tree heights. Contour lines are generated to match characteristics of target shown on the right. The ABSD similarity operator is used. White areas represent ocean. The land is colored using the mean of distributions. Blue regions indicate distributions with low mean. Note that pseudo-coloring based on the mean alone is not sufficient to identify regions with similar distribution patterns as the target.

Algorithm 1 Contour Line Interpolation

for all adjacent pairs of multi-value data items $M_1$ and $M_2$
if $t \in [\text{Sim}(M_1, T), \text{Sim}(M_2, T)]$ then
subdivide along $M_1$ and $M_2$ to obtain $n$ interpolated multi-value data items $I_1, I_2, \ldots, I_n$
$d_{m, n} \leftarrow +\infty$
for $i \leftarrow 1..n$
do $d \leftarrow \text{Sim}(I_i, T)$
if $d < d_{m, n}$ then
$d_{m, n} \leftarrow d$
s \leftarrow i
end if
end for
$s$ is the interpolation point between $M_1$ and $M_2$
end if
end for

Our lidar data set was collected over Alaska High Island and the topology of this region is illustrated in Figure 7. Contouring results on the lidar data are shown in Figure 8.

Because we are dealing with multi-value data sets, contour lines can mean different things. In one instance, when the multi-value 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, when the target itself is a multi-value data, then the contour lines are places where the multi-value data points are very similar to the target. One can devise further variations to depict additional properties of multi-value data sets as the need arises. For example, if one were to look for multi-value data sets whose pdf looks similar to the target in Figure 8(a) 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 Algorithm 2. $d$, $d_{m, n}$, and $d_{m, i}$ are distances computed using the similarity operator $\text{Sim}(M_1, M_2)$, and $\pm S$ is the range of mean offsets for $M_i$.

Isosurfaces

Isosurfaces are the 3D counterpart for contour lines. In Figure 9, we illustrate an isosurface generated by comparing each of the multi-value data in the volume against the multi-value target shown on the right. The data are the multi-value sound speed from the ocean data set. The KL similarity operator (Equation 7) is used 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 more than the 2D case, the isosurface shown in Figure 9 is actually an approximation. Unlike the case for 2D contours, a standard marching cubes algorithm is used on the scalar volume returned by comparing each multi-value 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 multi-value data points along an edge are close to each other. Of course, the same subdivision method as the one used in the 2D case could be used here as well.

Algorithm 2 Algorithm for Multi-value Matching with Shifting

$d_{m, n} \leftarrow +\infty$
for all distributions $M_i$ in the domain do
for $j \leftarrow -S..S$ do
shifted $T \leftarrow$ vary the mean of $T$ by $j$ intervals
$d \leftarrow \text{Sim}(M_i, \text{shifted}T)$
if $d < d_{m, n}$ then
$d_{m, n} \leftarrow d$
end if
end for
d_i \leftarrow d_{m, n}$
end for
Streamlines

Streamline generation is a standard flow visualization technique. They 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 below, is used to generate streamlines in multi-value velocity fields.

\[ P_{i+1} = P_i + \bar{v}(P_i) \Delta t \]  

(12)

where \( P_i \) is the current position at time step \( i \), \( \bar{v}(P_i) \) is the velocity at \( P_i \) and \( \Delta t \) is the integration time step.

Given a seed point \( P_0 \), the velocity at that location is a multi-value vector. That is, there are multiple possible trajectories resulting in different streamline paths. In the succeeding time steps, each of the trajectories will have their own set of multi-value velocities to deal with i.e., \( P_i \) and \( P_{i+1} \) are multi-value 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 multi-value datum \( M \) results in a multi-value datum \( M' \) where each member of the multi-value datum is multiplied by the scalar quantity: \( M' = sM \).

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

A slightly different interpretation is to apply the \( T_{ToScalar}() \) 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 multi-value position. A \( T_{ToScalar}() \) 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 \( \Delta t \)) to a multi-value data at each integration step. In this case, the seed point is first promoted to a multi-value location by simply replicating its values \( n \) times. This then requires an addition \( \oplus \) of two multi-value data which results in \( P_{i+1} \) being a multi-value location. The process is then repeated in subsequent iterations. The process of generating streamlines using convolution addition on the Massachusetts Bay data set is illustrated in Figure 10.

Because multi-value 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 multi-value 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 possible positions for each integration step. An additional parameter \( S \) may 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 11(a) shows the current practice of drawing "spaghetti plots" where streamlines are generated independently using each velocity field from the multi-value 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 and represents the mean streamline from the seed point. Figures 11(b) to (d) illustrate the appearance of multi-value streamlines rendered using transparent white circles as described previously. They convey the variability or uncertainty of the streamline as defined by the multi-value vector field.

Pathlines

Pathlines trace the path of a particle over a time varying flow field [2]. Again, using Euler integration for the sake of simplicity in illustrating how time varying multi-value velocity fields can be visualized, the relevant integration equation is:

\[ P_{i+1} = P_i + \bar{v}(P_i, t) \Delta t \]  

(13)

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.

A pathline from an ensemble of time varying flow data is illustrated in Figure 12. Pathlines can be rendered with semi-transparent circles in the same way streamlines are rendered (described in the
previous section). 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 conducted temporal interpolation of the ensemble flow field. For the example in Figure 12, we introduced 30 additional time frames within each three-hour forecast period using the interpolation strategy for multi-value data items described in Equation 9. This procedure produced an ensemble velocity field every 0.1 hours, allowing us to use the same integration step in our simple fixed step Euler integration. Binwise addition was used. 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. Note that 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.

5 Discussion and Conclusion

We introduced the multi-value data type and three different approaches for visualizing spatial multi-valued data sets. The parametric approach is the traditional visualization method offering, well accepted statistical reasoning but limited primarily to 2D domains with well-behaved distributions. The shape descriptor approach is also limited primarily to 2D but allows visualization of multimodal distributions. One could continue designing visualization techniques specialized for dealing with multi-value data sets. Instead, the approach we took was to find a systematic way in which existing visualization techniques could be adapted to work with multi-value data sets directly. This resulted in the operator based approach. This approach allows us to draw upon different fields to define operators appropriate for the data set, and visualization task at hand. It also addresses the shortcomings of the two other approaches namely flexibility in handling both parametric and non-parametric distributions and the ability to go beyond 2D spatial multi-value data sets.

Using the operator based approach, we demonstrated how to convert workhorse visualization techniques such as pseudo-coloring, contour lines, isosurfaces, streamlines and pathlines to work directly with multi-value data sets. Aside from these, there is other interesting work that needs to be done. Some of the extensions we are looking at include: volume rendering multi-value data sets, interpolation of spatially sparse multi-value data, and feature extraction of multi-value data sets. The operator based approach is particularly important when dealing with large time varying multi-dimensional, multivariate, multi-value data sets as they are a factor n times larger than their single value counterpart.

The three approaches are listed in the order of their development as well as generality. While the operator based approach is the most flexible and powerful approach for handling multi-value data, it also requires additional training in its use and interpretation.
A simple example illustrated how the interpretation of contour lines and isosurfaces needs to be modified to mean similarity to a target distribution as opposed to matching a threshold value. Others, such as streamlines, force the users to acknowledge the inherent uncertainty or variability captured by the multi-value field. Another important factor that affects the interpretation of the results is the use of the underlying operators. We illustrated the difference between binwise and convolution additions. Other types of additions can conceivably be constructed which would also require the users to know their nuances and assumptions.

So, how does one choose among the multitude of alternatives? Obviously, there 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 or not. Even with these filters to reduce the number of choices, one may 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, one 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, etc. In summary, while the operator based approach is powerful and flexible, it does require some sophistication and care from the user.

6 Acknowledgment

This work was 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. Finally, we would like to thank the reviewers for their helpful comments to help improve the flow and readability of the paper.

References


