VISUALIZATION TECHNIQUES FOR SPATIAL PROBABILITY DENSITY FUNCTION DATA

***Udeepta D. Bordoloi*[1]*, David L. Kao*[2] *and Han-Wei Shen*[1]**

[1]*Dept. of Computer and Information Sciences, The Ohio State University, Columbus, OH, USA*
Email:{bordoloi,hwshen}@cis.ohio-state.edu

[2]*NASA Ames Research Center, Moffet Field, CA, USA*
Email:davidkao@nas.nasa.gov

ABSTRACT

Novel visualization methods are presented for spatial probability density function data. These are spatial datasets, where each pixel is a random variable, and has multiple samples which are the results of experiments on that random variable. We use clustering as a means to reduce the information contained in these datasets; and present two different ways of interpreting and clustering the data. The clustering methods are used on two datasets, and the results are discussed with the help of visualization techniques designed for the spatial probability data.

Keywords: Spatial data, uncertainty, probability density functions, visualization, clustering

1 INTRODUCTION

Probability based techniques are being used in increasingly diverse fields. With more and more probabilistic data being generated every day, it becomes imperative that the users of the data have adequate tools for data exploration and analysis. Along with statistical approaches, visualization techniques can provide unique insight to the data, in particular for the purpose of revealing the spatial arrangement and distribution of the underlying dataset. Proper visualization tools can also help the user browse through the data and perform interactive queries. This is especially useful when dealing with *spatial probability density function* datasets, which are the focus of this paper.

The term “spatial probability density function dataset” is used in this paper to represent a spatial (2D or 3D) dataset of one-dimensional probability density functions (*pdf*s). The dataset is defined over a 2D or 3D region, just like any other spatial dataset. However, each pixel (x,y) represents one random variable **v**(x,y). We are interested in visualizing the pdfs of the random variables: ***f***(**v**(x,y)). Please note that in some literature, the words ‘spatial uncertainty’ or ‘spatial probability’ are used with an alternative meaning. They are used to denote the probability that an event will occur at a given location. For example, a pdf ***g***(x,y) might give the probability of a certain event occurring at the position (x,y). In contrast, the pdf we are concerned with, ***f***(**v**), gives the probabilities of the variable **v** taking different scalar values, and there is one such random variable at each pixel (x,y) in our datasets. Scientists studying these datasets are interested in viewing the probability density functions at specific locations, and also in understanding the collective behavior of the random variables encompassed within a larger region. Sometimes, instead of the pdf ***f***(**v**), the dataset consists of samples of the variable **v**, from which the pdf ***f***(**v**) can be approximated (Silverman, 1986). In this case, these probabilistic spatial datasets can also be viewed as multiple instances of spatial scalar data, where each instance contains one sample for (the random variable at) each pixel in the entire spatial domain. That is, each instance can be thought of as a spatial dataset of scalars. In this paper, we present visualizations for both the interpretations: first we look at multiple instances of spatial scalar data (Section 4.1), and then we visualize the same dataset as a single spatial dataset of pdfs (Section 4.2).

For any single pixel, an estimate of the pdf of the associated random variable can be obtained from the samples using different density estimation techniques (Silverman, 1986), and visualized by simple 2D graphing techniques. As the number of pixels increases, a simultaneous visualization of all the graphs becomes progressively more difficult. For a spatial (2D/3D regular grid) pdf data, visualization of probability density functions for all the pixels in the spatial domain in a single picture becomes intractable. Even if one were to discount the limited drawing space and allow for multiple screens, it would be very laboring for the user to go through each individual graph. To resolve this issue, in this paper we propose techniques that can effectively summarize the data and present only the salient information about the underlying probability density functions. Our techniques are based on *clustering*, which is used as a means to group together similar data. All the probability information within one group (cluster) is summarized to produce a more comprehensible visualization. As we have mentioned, a probabilistic dataset can be thought of as a spatial dataset of random variables, and alternatively as multiple instances of spatial scalar data. Each way of interpreting the data leads to a different form of clustering. For the first case, pixels in the spatial domain are grouped together based on the data distribution of the random variable. In the latter case, the different instances of the dataset are divided into several groups. The instances will be similar to each other within each group, and different from the ones in other groups.

Based on our clustering framework, we introduce novel visualization techniques to aid scientists in exploring and understanding spatial probability density function data. Interactive browsing is extremely important for efficient visualization; hence, a hierarchical clustering technique is used which allows the user to control the amount of information he/she wants to visualize. If the user is interested in a particular region and wants more detail, he/she can interactively reduce the size of clusters representing the region by splitting the clusters into their children. When clustering pixels, we use a cluster 'average' based coloring scheme which allows intuitive split/unsplit operations on the hierarchical clustering results. We also introduce procedural patterns as a technique to visualize various statistical properties of the spatial clusters. We show examples where the patterns help the user to visually distinguish between clusters with similar but not exactly same statistical properties. The different visualizations allow the user to study not only the individual probability densities but also the collective behavior of the variables.

2 RELATED WORK

Research associated with the many different types of distribution data has mostly been confined to the particular scientific communities which have generated the data. But the problem of visualization and exploration of the general data type of probabilistic spatial data has remained under-explored. Below, we will mention briefly the available research in a few closely related topics such as visualization and exploration of uncertainty datasets.

Visualization of uncertainty data has inspired a broad variety of solutions (Pang, Wittenbrink & Lodha, 1997; Wittenbrink, Pang & Lodha, 1996; Lodha, Wilson & Sheehan, 1996a; Minghim & Forrest, 1995; Cedilnik & Rheingans, 2000) (Barnhill, Opitz & Pottmann, 1992) (Wittenbrink, 1995) (Lodha, Sheehan, Pang & Wittenbrink, 1996b) (Grigoryan & Rheingans, 2002). Most of these methods consider uncertainty to be a scalar quantity. At each data location, two values are extracted: the expected value of the data (mean) and the error in the data, which might be approximated by, say, the variance. While the expected value and variance are two important statistical properties, the pdf datasets contain much more information than just the mean and variance. An example of distribution visualization is provided by Ehlschlaeger, Shortridge & Goodchild (1997). Animations of the different instances of spatial data are used to help the user gain insight to the uncertainty at the different pixels. Kao, Dungan & Pang (2001) have developed techniques to study probability density functions (pdfs) in greater detail. They estimate the pdfs from the samples in the probability datasets, and then use various statistical summaries of the pdfs, e.g., mean, variance, skewness, kurtosis, inter-quartile distance etc. to construct a dense global visualization of dataset. The user has to probe each point to visualize the estimated pdf. Figure 1 shows a visualization from (Kao et al, 2001). The lower layer shows a colormap of the mean of the Landsat dataset described in Section 3. The upper layer consists of a surface representation where the height of the surface signifies the standard deviation at each pixel. The colormap for the surface is obtained from the interquartile distances of the pdfs at each pixel. The vertical bars signify the absolute value of difference between the mean and the median at each pixel. As can be seen, the visualization can get quite cluttered. Kao, Dungan & Pang (2002) present additional methods of visualizing 2D pdf data using density estimate volume visualization and shape-based descriptors for the pdfs. In contrast, our hierarchical spatial clustering gives a multi-resolution representation of the distribution data, and the user can interactively visualize a representative distribution or the pdfs for each cluster at various levels of detail.

The whole literature of clustering is too wide to mention here, so we will only mention some of the research that is directly related to our work. Some of the popular clustering algorithms are K-means, Pam (Kaufman & Rousseeuw, 1990), Clarans (Ng & Han, 1994), DBScan (Ester, Kriegel, Sander & Xu, 1996), Cure (Guha, Rastogi & Shim, 1998), Rock (Guha, Rastogi & Shim, 1999) and Chameleon (Karypis, Han & Kumar, 1999). Clustering has been used previously to aid in visualization of different types of data on spatial domains (Heckel, Weber, Hamman & Joy, 1999) (Telea & van Wijk, 1999) (Tilton & Lawrence, 2000). Any clustering algorithm can be used for our purposes; however hierarchical clustering is better for interactive visualization as it allows the user to view the clustering results in different levels of detail.

3 DATASETS

We present our visualization techniques with two spatial datasets. In these datasets, each pixel or voxel is a random variable, and contains multiple samples. The samples can be used to estimate the probability density function of that variable. For the sake of simplicity, we will assume a 2D dataset for our discussions, and the terms 'pixel' and 'image' will be used. However, the techniques work for 3D data too, and the terms can be replaced with 'voxel' and 'volume'.

These probability datasets were generated by computer simulations (which are analogous to *experiments* for a random variable). Each simulation was run for all the pixels simultaneously, as opposed to running the simulations for each pixel independently. This is because (the random variables at) the different pixels are not independent. In fact, neighboring pixels usually have similar-valued samples, that is, spatial autocorrelation tends to be high. The result of each simulation is an image, which we will refer to as a single *realization* or a single instance. Each realization represents a possible outcome for the spatial collection of random variables. Our datasets consist of a finite number of such realizations. These datasets do not actually store the pdfs of the pixels. Instead, we have samples for each pixel, with one sample coming from one realization. The samples can then be used to calculate different statistical properties such as mean and variance for each pixel. There are also many different density estimation techniques to find the (unknown) pdf at each pixel. For example, histogram is a crude and fast method of approximation. A more accurate estimate of the pdf can be obtained using the kernel estimator (Silverman, 1986) (Kao et al, 2002).

Our first dataset is constructed from a small region in the Netherlands imaged by the Landsat Thematic Mapper (Dungan, 1999). For this dataset, the biophysical variable mapped across this region represents percent forest-cover. 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 are assumed to be available. This spectral vegetation index is related to forest cover in a linear fashion but with significant unexplained variance. The ground area represented by a field measurement is assumed to be equal to the area represented by one pixel. A set of realizations was generated by conditional co-simulation (Deutsch & Journel, 1998) using both ground measurements and the coincident satellite image. The data set consists of 101 x 101 pixels and 250 realizations. Values range from 0 to 255, re-scaled from percentage forest-cover (Kao et al, 2002). A visualization of this dataset from (Kao et al, 2001) is shown in Figure 1. Different clustering results for this dataset are shown in Figure 3.

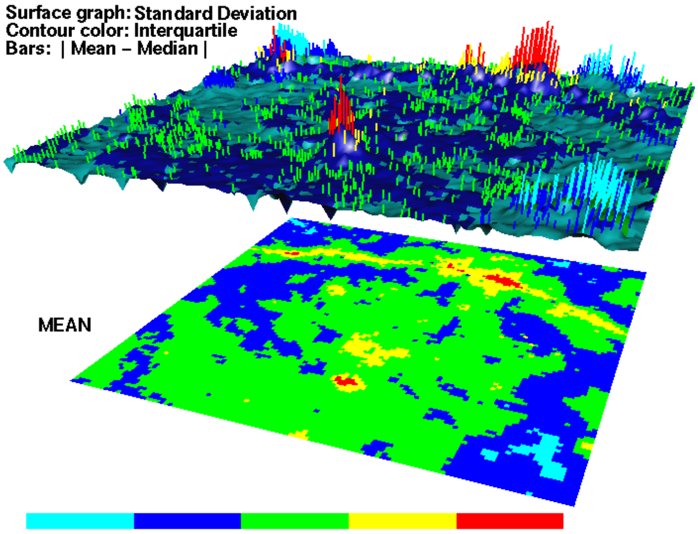

Surface graph: Standard Deviation
Contour color: Interquartile
Bars: | Mean – Median |
MEAN

Figure 1. Pixel-wise visualization of the Landsat dataset from (Kao et al, 2001). The dataset is described in Section 3. The lower layer shows a colormap of the mean. The upper layer denotes the standard deviation, interquartile distance, and the absolute value of the difference between mean and median at each pixel. Cyan denotes low mean forest-cover and red denotes a high-mean.

Our second probability dataset is from an ocean model covering 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 with fields such as temperature and salinity. To dynamically evolve the physical uncertainty, an Error Subspace Statistical Estimation (ESSE) scheme (Lermusiaux, 1999) 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. We currently have access to the Monte-Carlo forecasts of the 3D volume for a single instant in time. That is, we have multiple realizations of the 3D volume for one time-step. The field value is for sound speed and is derived from the other physical field values. The dimension of this dataset is 65 x 72 x 42, with 80 realizations of the volume. Different realizations of the top-most layer are shown in Figure 2, and different 2D slices of a 3D clustering of this dataset are shown in Figure 4.

These spatial datasets lie in 2D (or 3D) physical space. If all the realizations of the 2D (or 3D) dataset are stacked on top of each other, it will result in a 3D (or 4D) dataset. Although it is possible to visualize these stacked datasets using traditional techniques such as volume rendering (Lichtenbelt, Crane & Naqvi, 1998), the visualizations are unintuitive and difficult to interpret. One of the reasons for this is that the realizations are arranged in no particular order. They are numbered arbitrarily, and all numberings are equivalent. Please note, however, that this arbitrary numbering of realizations is the same for all the pixels, i.e., the first sample of each pixel comes from the same realization, and so on.

4 CLUSTERING

To enable easy visualization and interaction with the data, we use clustering algorithms. As we have mentioned before, the datasets presented in the previous section can be viewed in two different ways. They can be thought of as: (i) spatial datasets with a pdf at each pixel, or (ii) a set of instances of spatial datasets of scalars, each instance representing a possibility for the dataset. These two interpretations of the data lead to two different types of clustering. The data can be clustered along the spatial dimensions by finding groups of pixels which have the same pdfs. Or, different realizations can be grouped together based on how similar they are. Both the types of clustering generate different kinds of information for the user. Spatial clustering of the datasets will partition the space into regions such that within each region pdfs of the pixels remain relatively unchanged. By reducing the large number of pixels into a few regions, this helps the user gain an understanding of the behaviors of different areas of the dataset. The user can now also study how the behavior of one region compares to that of another. As an alternative, if we cluster the different instances (realizations) of the same dataset, it will yield a few groups of realizations. By ignoring the small variations within the groups, each group can be visualized as one distinct possible outcome for the simulation process. Since the number of possible outcomes the user has to deal with has been reduced, it becomes much easier for the user to understand the different possibilities. The number of realizations within each group will approximate the likelihood of the dataset having the outcome represented by the group. Groups consisting of a very small number of realizations can be considered to be outliers and ignored, or they can be considered as special cases, depending on the situation. The group with the greatest number of realizations will be the most likely case.

Overall, our clustering based approach possesses three main advantages for analyzing spatial distribution datasets:

1. **Information Summary**: By presenting a high level view of the dataset, clustering reduces the amount of human effort required to explore and understand the probabilistic behavior. The hierarchical nature of the clustering allows the user to browse through the datasets in different levels of detail, as dictated by the constraints of precision and effort. In the case of spatial clustering, instead of probing each pixel for its realization, the user can probe a cluster of pixels. A cluster 'representative' probability density gives an approximation to the individual densities at each pixel within the cluster. By clustering the realizations, the user can view the representative instances for each cluster. After clustering, the user does not have to sift through all the data manually; it is much easier for him/her to view information from the clustered data.

2. **Feature Extraction**: In spatial clustering, the cluster shapes and sizes capture spatial structures and patterns which help in the understanding of the underlying phenomena. We show examples where clusters bring out structures expected to be present in the datasets. Domain scientists are interested in the shape and location of structures like the arc (called a 'road') in Figure 3 and the middle band (called a 'shelfbreak') in Figures 2 and 4.

3. **Visualization**: Clustering allows us to incorporate various statistical properties of the spatial structures (like mean, variance, skewness) to improve the visualizations. A quick comparison between Figures 1 and 3 will show the difference between the pixel-wise visualization approach and the clustering approach. Figure 1 is a pixel based visualization of the Landsat dataset from (Kao et al, 2001), while Figure 3 shows clustering results of the same data.

In the following sections, we present the details of the two different clustering schemes that we propose.

4.1 REALIZATION CLUSTERING

We will discuss the clustering of the different realizations using the shelfbreak spatial uncertainty dataset. We use the top-most slice of the 3D volumetric realization dataset. So, the data input for clustering is 80 realizations of 65 x 72 spatial scalar data. Each 2D realization can be thought of as an image; hence, this translates to a clustering of different images. We can use techniques from the existing literature for image comparison to perform distance tests between the realizations. The literature for image similarity metrics is quite extensive. Popular methods lie in the class of histogram matching, texture analysis, feature matching, and multi-resolution methods (Jacobs, Finkelstein & Salesin, 1995; Santini & Jain, 1999; Chi Wong, Bern & Goldberg, 2002). We have used an image matching scheme based on wavelets.

4.1.1 DISTANCE FUNCTION FOR CLUSTERING

We want to be able to ignore minute differences between realizations, and cluster based only on the large dissimilarities. A Haar wavelet transform is used to create a multi-resolution representation of each realization. The highest frequency components are discarded to get a low frequency representation, which discards the tiny variations in the data, and also reduces the effect of noise. Next, the variance of each coefficient is calculated over all the realizations. The coefficients with very small variance are also discarded, as they do not contain much information that is useful in discriminating the realizations. This reduces the processing required for the computationally expensive clustering process. For each realization R, the remaining wavelet coefficients are stacked together to form a signature vector **r**. If *n* wavelet coefficients are chosen to form the signature, then

$$\mathbf{r} = [r_1 \quad r_2 \quad \cdots \quad r_n] \text{ , where } r_i \text{ is a wavelet coefficient of realization R.}$$

We define the distance between two realizations as the sum of absolute value of the differences between corresponding coefficients of their signature vectors, i.e., the Manhattan distance between the two vectors.

$$dist(R,S) = \sum_{i=1}^{n} | r_i - s_i | \text{ , where } r_i \text{ and } s_i \text{ are the wavelet coefficients of realizations } R \text{ and } S.$$

The comparison method presented here finds large scale differences between realizations, while ignoring the small differences. Thus realizations with similar global structure will be grouped together. Small variations will be ignored.

4.1.2 CLUSTERING

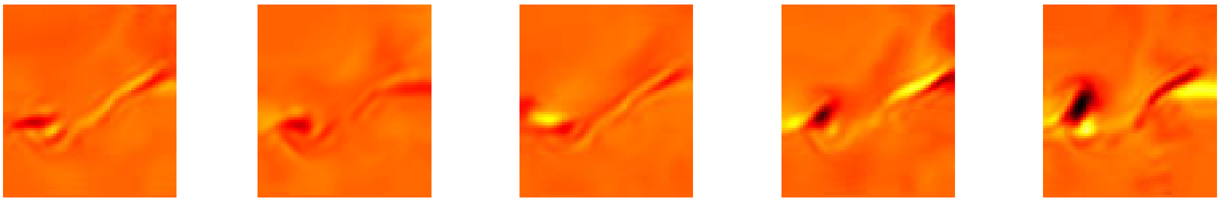
A bottom-up clustering technique has been used for the results presented in this paper. Clusters are merged progressively, which creates a tree of clusters, also known as a *dendrogram*. At first, each realization is defined to be a cluster by itself. We then find and merge the two neighboring clusters such that the resulting parent cluster has the least *error* of all possible mergers. We define the error of a cluster as the maximum distance between any two realizations within that cluster. The error of a cluster Θ is the maximum distance between any two realizations *R* and *S* within the cluster.

$$error(\Theta) = \max(dist(R,S)) \text{ , where } R, S \in \Theta \text{ .}$$

The clustering process is continued iteratively till we are left with a single cluster that contains all the realizations in the dataset. This last cluster is the root of the dendrogram, which is the end result of the clustering algorithm. The clustering algorithm presented here is similar to the *complete linkage* method, also known as the farthest neighbor method (Manly, 1994). Since we define the error of a cluster as the greatest distance between two member realizations of the cluster, this method tends to create tight clusters whose members are very similar to each other. This method was chosen because the tight clusters imply that the representative realizations would be very close to the actual realizations within the clusters. To restrict the size of the cluster-tree, we maintain clusters only up to a user-specified level of tree. In general, every node in the tree (except the leaf nodes) will have two children. Further reduction of the tree size can be obtained by throwing away alternate levels of the tree; each node will then have four children. The clustering is done as a preprocessing step. At run-time, the results of clustering (in the form of a dendrogram) are read in and used for interactive exploration. Figure 2 shows the results of clustering the 80 realizations of the shelfbreak dataset. Three of the largest clusters are shown. None of the larger clusters have realizations with the dark region that can be seen in the two outliers. This implies that it is highly unlikely that the dark region will be present.

4.2 SPATIAL CLUSTERING

For our second visualization method, we use hierarchical clustering techniques to cluster the spatial domain into different regions. This method exploits the spatial autocorrelation present in the spatial datasets. Neighboring pixels tend to have very similar statistical properties and similar pdfs, hence it leads very naturally to the idea of grouping together pixels which are similar to each other. The resulting groups are then used to create a compact representation of the pdf dataset. The pixels in each region will have similar pdfs, and instead of visualizing the individual pdfs (or other statistical properties), we can show the properties of the region. The locations, shapes and sizes of the regions are all useful information borne out by the clustering process, and are very important in understanding the underlying phenomena.



8(10%) 8(10%) 7(8.75%) Outlier Outlier

Figure 2. Results of realization clustering for the shelfbreak dataset. From the three largest clusters, a single member realization is shown. Also shown are two clusters with single realizations, which can be classified as outliers. The number of realizations in the clusters is shown below the images. The ratio of the number of cluster members to the total number of realizations is expressed as a percentage value within the parentheses.

4.2.1 DISTANCE FUNCTION FOR CLUSTERING

One possibility of clustering the pixels is using their pdfs. As mentioned before, the pdf at each pixel can be constructed from the realizations by using density estimation methods, e.g., histograms, kernel estimator etc. (Silverman, 1986) (Kao et al, 2002). Once the pdfs are generated, we can compare the distance between two pdfs using any of the established metrics: Kullback-Leibler distance, entropy etc.

This is, however, not the best way to evaluate the distance function for our datasets. These datasets contain realizations, which contain more information than just the pdfs at each pixel. There is information about how the pixels behave together, i.e., correlation between pixel values (spatial autocorrelation). This is the same information that is indirectly used in realization clustering (Section 4.1). A distance metric based on estimated pdfs assumes each pixel to be an independent random variable, and thus discards the correlation information. We present a simple example which shows the misleading effects of using pdfs for clustering. Consider two random variables X and Y that take values in the range [0, 1] with uniform probabilities. Also, suppose Y is always equal to (1-X). Both X and Y have the exact same pdf, and will be clustered together if pdf is used for comparison. However, such a clustering is very misleading because X and Y behave in opposing ways. In spatial datasets, there is usually a significant correlation between neighboring pixel values. For the datasets that we use in this paper, we have found that the correlation between neighboring pixels ranges from 1.0 to -1.0. This implies that there exist neighboring pixel pairs with very high positive and negative correlations.

Instead, we use a correlation preserving distance function based on the realization values of each pixel. When comparing two pixels, we match the pixel values from the corresponding realizations. For each pixel P, we stack all its realizations to form a *realization vector* **p**. If there are *n* realizations in the dataset, then

$\mathbf{p} = [p_1 \quad p_2 \quad \cdots \quad p_n]$, where p_i is the value of the *i*-th realization of pixel P.

The distance between two pixels *P* and *Q* is defined as the Manhattan distance between their realization vectors.

$dist(P,Q) = \sum_{i=1}^{n} | p_i - q_i |$, where p_i and q_i are the values of the *i*-th realizations of pixels *P* and *Q*.

The Manhattan distance is computationally inexpensive and has yielded satisfactory results for both the datasets.

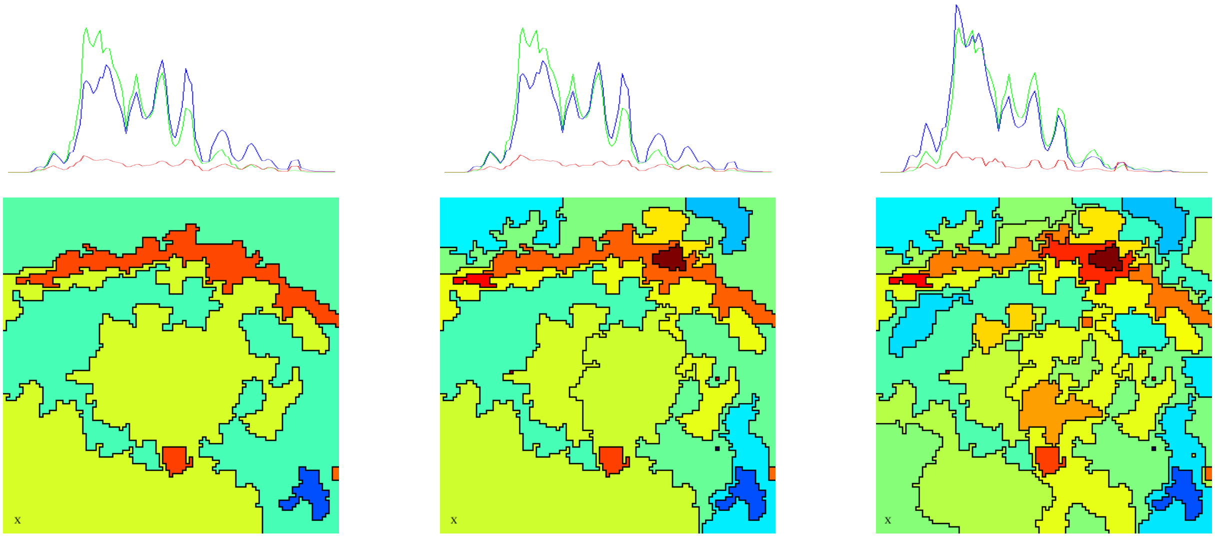
4.2.2 CLUSTERING

As in realization clustering, we use a bottom-up hierarchical clustering method. It is desirable to have clustering results which are easy to visualize, so we add the constraint that a cluster cannot be formed from two non-contiguous regions. This type of clustering (with a contiguity constraint) is also known as regionalization in some literature.

Initially, each pixel is a cluster by itself. Clusters are progressively merged together using a greedy algorithm. Only those pairs of clusters which are 4-neighbors are candidates for merger. The pair which will result in the least merging error is selected and grouped together into a single cluster. After each merger, the neighbor information is updated. Like Section 4.1.2, the error of merging is defined as the error of the parent cluster produced. We use the complete linkage (farthest neighbor) definition of error. The error of a cluster Θ is the maximum distance between any two pixels *P* and *Q* within the cluster.

$error(\Theta) = \max(dist(P,Q))$, where $P, Q \in \Theta$.

The process is repeated till we are left with a single cluster, and the result of the whole process is stored in a dendrogram. The clustering results can be visualized at different levels of the dendrogram. In Figure 3, we show the clustering results for the Landsat dataset.



(a) Coarse clusters (b) Medium clusters (c) Fine Clusters

Figure 3. Hierarchical clustering of the Landsat dataset. The middle row shows three different levels of clustering, from coarse to fine. The clusters are colored by their mean values. The top row shows the pdf for the point 'X' in the lower left corner of the dataset (green curve), and the pdf for the cluster that contains the point 'X' (blue curve).

4.3 VISUALIZATION

Proper interaction interfaces can add a lot of value to the results of the clustering techniques discussed in the previous sections. The users need to browse through the different clustering levels to arrive at the clusters they are satisfied with. The clustering tree (dendrogram) can be cut at different levels by changing the error threshold. The different thresholds yield different clustering results, and the user can study the merging/splitting of clusters in an interactive manner. The user can probe the pdf at each pixel, and also visualize the 'average' pdf of each cluster. The 'average' pdf is estimated by assuming that the cluster is represented by a single random variable, and assigning all the realizations of all the pixels within the cluster to that variable. As the cluster size decreases, the cluster 'average' pdf starts to match the individual pdfs more and more.

The clusters can also be split/unsplit manually by clicking on them. To help the user keep a sense of continuity in this manual process, it is desirable that the children clusters (in the dendrogram) bear some resemblance in color to the parent cluster, and also to each other. Also, clusters which are not related should have colors that are not too similar. We have tested various procedural color assignment techniques in different color spaces, but without encouraging results. As the number of colors increases, it is difficult to assign colors that satisfy both of the conditions mentioned above. Therefore, we decided to relax the second condition and allow clusters not related to each other in the dendrogram to have similar colors. We use a color scheme in which the color of a cluster is determined by its mean. Since means of sibling clusters are similar to each other and also to the parent, this gives us a nice solution which satisfies the first condition. Moreover, the colors give some information about a statistical property of the cluster. This scheme is used for the visualizations shown in Figure 3. The spatial structure of the dataset is extracted by the spatial clustering, and the arc-like spatial structure (called a 'road') stands out because its average value is quite different from that of the surrounding region.

In this scheme, it might happen that the visual impact is weakened by rendering two neighboring clusters with possibly very similar colors. We solve this issue by the use of patterns which can provide information about different statistical summaries of the clusters, and thus highlight the differences between the two neighboring clusters. Statistical summaries such as the mean, variance, skewness, kurtosis, and inter-quartile range contain information about various properties of the pdfs. Skewness is a measure of asymmetry in the tail of the pdf, kurtosis signifies how flat a pdf is, and the inter-quartile distance gives an idea about the spread of the pdf. Examples of using such patterns are shown in Figure 4, which show 2D slices of the 3D clustering results for the 65 x 72 x 42 shelfbreak dataset. The clustering is done over a 3D domain, and is a straightforward extension of the 2D clustering presented in the previous section on spatial clustering. Each image shows the visualization of a 2D slice (at a fixed depth) of the 3D clustered volume. In this example, we have used variance and skewness as the two inputs for the patterns in the clusters. The variance is normalized to a range of [0,1] for convenience. Skewness values are scaled by a constant so that all clusters have skewness in the range [-1,1]. The uniform mean color of clusters is now replaced by alternating lighter and darker band patterns. The width of the darker band in each cluster is directly proportional to the variance within that cluster. The color of the lighter band comes from the mean. The orientation of the darker lines represents the third statistical variable (skewness). If the skewness of a cluster is zero, the lines are vertical. We rotate the patterns clockwise for positive values or counter-clockwise for negative skewness. The angle of rotation is directly proportional to the skewness within the cluster. The maximum possible rotation (for values of -1 or 1) is a little less than ninety degrees. From these figures, we can see that most of the randomness is present along the middle region in the slices with lesser depths. This part is called the 'shelfbreak', and is the region of interest in the dataset. The large yellow cluster has very little variance compared to other clusters. Most of the clusters have negative skewness. There are two green clusters towards the right edge of the ocean dataset, which would have been very similar to each other but for their opposing skewness. This is an example where the patterns representing additional statistical summaries have helped differentiate two clusters with similar means.

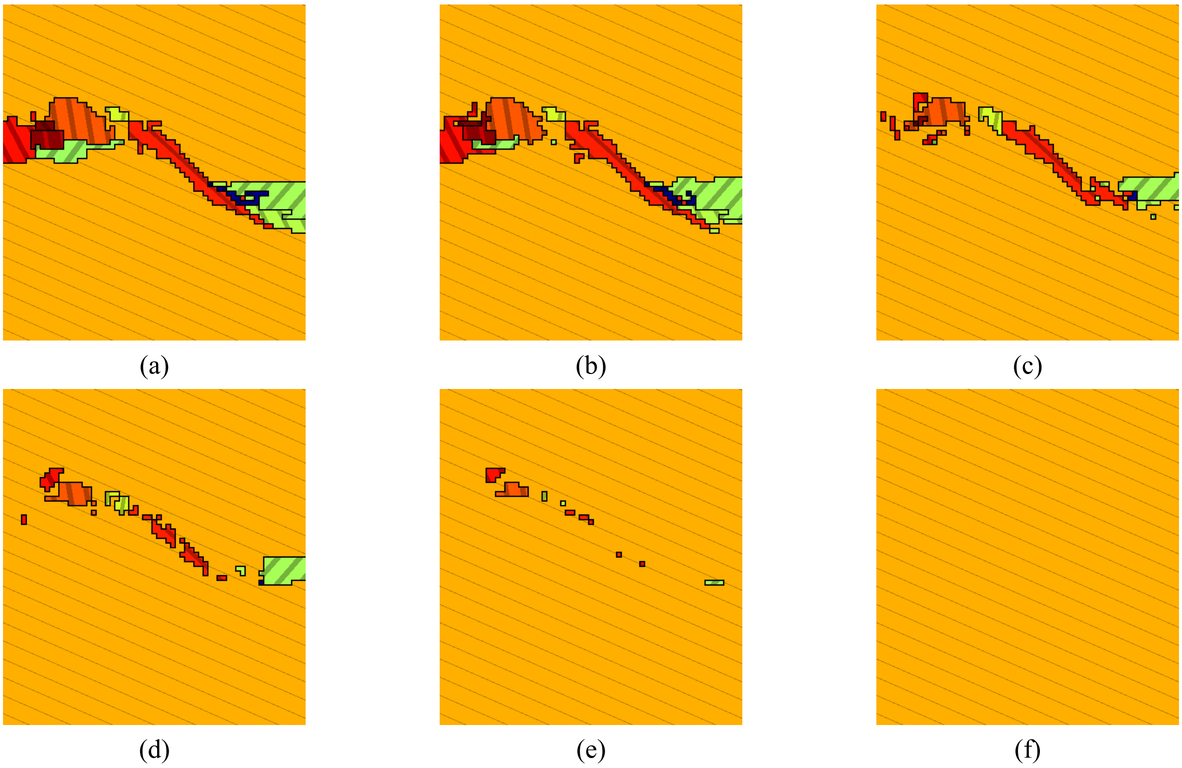


Figure 4. Visualization of 3D clustering results for the shelfbreak dataset. The images show the clustering results at different depths. The region of the shelfbreak is clearly visible from the clustering results. As the depth increases, the structure begins to break down, and finally disappears.

5 CONCLUSION

We have presented two different methods of clustering spatial pdf data – clustering of realizations, and spatial clustering. Each technique yields unique and useful information about the data. Based on what kind of knowledge the user hopes to derive from the cluster analysis, the user can choose either one or both.

Many probability datasets have a large number of realizations, making it difficult for a human to study each one individually, and to compare each one with others. Also, while making comparisons, it is useful to be able to ignore the small variations, yet note the large differences present in the realizations. The human effort required for these tasks can be reduced with the help of a computer aided technique such as realization clustering. The clustering process not only reduces the information presented to the user, but also finds outliers in the data. Since features in the data (such as peaks, valleys etc.) remain relatively unchanged within a cluster, comparing them across clusters also helps in visualizing the uncertainty in their location, size and shape. The utility of realization clustering begins to reduce if all realizations in the data are very similar to each other, or if all are almost equally dissimilar to each other. In such cases, it will result in very similar looking clusters, or clusters with large variations within their own members. This is the case for the Landsat dataset, where the differences among the realizations are limited to fine details, and all the realizations look very similar at a coarse level of detail.

The spatial clustering allows users to get a global view of the spatial structures in the datasets. Without the use of clustering, it is very difficult for a human to figure out the different regions based on statistical properties. (Compare Figure 1 with Figure 3). Since all the pixels in a cluster have similar probabilistic behavior, the number of pdfs that the user needs to study is reduced dramatically. While the clustering brings out various features of the datasets, it is not a substitute for domain specific feature extraction methods, where domain knowledge about the data can be used.

Together with the visualization techniques presented, the spatial clustering method helps the users with improved interactivity and ease of understanding the spatial pdf data. Since a hierarchical clustering scheme is used, the user can view the datasets at any desired level of detail. The user is also able to selectively increase the detail of one region by going down the cluster tree for that region, while other regions remain at a coarse level. Thus both detail and context are presented in the visualization. For large datasets, the screen space can prove to be a limiting factor in interactivity, specially when using detailed views. In the future, we hope to add focus and context type visualization tools to alleviate this problem.

6 ACKNOWLEDGEMENTS

We would like to thank P. F. J. Lermusiaux for the ocean dataset and J. L. Dungan for the Landsat dataset. A word of thanks also goes to J. L. Dungan, Alex Pang and Alison Luo for helpful discussions on the subject. We greatly appreciate the valuable feedback of the anonymous reviewers. This work was supported by NASA Intelligent Systems Program Cooperative Agreement NCC2-1261 and by NSF Career Award CCF-0346883.

7 REFERENCES

Barnhill, R.E., & Opitz, K., & Pottmann, H., (1992) Fat surfaces: A trivariate approach to triangle-based interpolation on surfaces. *Computer Aided Geometric Design 9*(5), 365-378.

Cedilnik, A., & Rheingans, P., (2000) Procedural annotation of uncertain information. *Proc. of IEEE Visualization 2000* (pp. 77-83). Salt Lake City, USA.

Chi Wong, H., Bern, M., & Goldberg, D., (2002) An image signature for any kind of image. *Proc. of International Conference on Image Processing 2002* (Vol. 1, pp. I-409-I-412). Rochester, USA.

Deutsch, C.V., & Journel, A.G., (1998) *GSLIB: Geostatistical Software Library*. New York: Oxford University Press.

Dungan, J.L., (1999) Conditional simulation: An alternative to estimation for achieving mapping objectives. In Meer, F., Stein, A., & Gorte, B., (Eds.), *Spatial Statistics for Remote Sensing* (pp. 135-152). Dordrecht: Kluwer Academic.

Ehlschlaeger, C.R., & Shortridge, A.M., & Goodchild, M.F., (1997) Visualizing spatial data uncertainty using animation. *Computers and GeoSciences 23*(4), 387-395.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X., (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. *Proc. of the Second Int'l Conference on Knowledge Discovery and Data Mining* (pp. 226-231). Portland, USA.

Grigoryan, G., & Rheingans, P., (2002) Probabilistic Surfaces: Point Based Primitives to Show Surface Uncertainty. *Proc. of IEEE Visualization 2002* (pp. 147-154). Boston, USA.

Guha, S., Rastogi, R., & Shim, K., (1998) CURE: An efficient clustering algorithm for large databases. *Proc. of 1998 ACM-SIGMOD Int. Conf. on Management of Data* (pp. 73-84). Seattle, USA.

Guha, S., Rastogi, R., & Shim, K., (1999) ROCK: a robust clustering algorithm for categorical attributes. *Proc. of the 15th Int'l Conf. on Data Eng* (pp. 512-521). Sydney, Australia.

Heckel, B., & Weber, G., & Hamann, B., & Joy, K., (1999) Construction of Vector Field Hierarchies. *Proc. of IEEE Visualization 1999* (pp. 19-25). San Francisco, USA.

Jacobs, C., Finkelstein, A., & Salesin, D.H., (1995) Fast Multiresolution image querying. *Proc. of SIGGRAPH 95*, in *Computer Graphics* Proceedings, Annual Conference Series (pp. 277-286). New York: ACM.

Kao, D., Dungan, J., & Pang, A., (2001) Visualizing 2D Probability Distributions from EOS Satellite Image-Derived Data Sets: A Case Study. *Proc. of IEEE Visualization 2001* (pp 457-460). San Diego, USA.

Kao, D., Luo, A., Dungan, J., & Pang, A., (2002) Visualizing Spatially Varying Distribution Data. *Proc. of Information Visualization 2002* (pp 219-225). London, UK.

Karypis, G., Han, E.-H., & Kumar, V., (1999) CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling, *IEEE Computer: Special Issue on Data Analysis and Mining 32*(8), 68-75.

Kaufman, L., & Rousseeuw, P., (1990) *Finding Groups in Data: an Introduction to Cluster Analysis*. New York: John Wiley & Sons, Inc.

Lermusiaux, P.F.J., (1999) 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.

Lichtenbelt, B., Crane, R., & Naqvi, S., (1998) *Introduction to Volume Rendering*, Upper Saddle River. USA: Prentice Hall, Inc.

Lodha, S.K., Wilson, C.M., & Sheehan, R.E., (1996a) LISTEN: sounding uncertainty visualization. *Proc. of IEEE Visualization 1996* (189-198). San Francisco, USA.

Lodha, S.K., Sheehan, R.E., & Pang, A., & Wittenbrink, C.M., (1996b) Visualizing geometric uncertainty of surface interpolants. *Proc. of Graphics Interface 1996* (pp. 238-245). Toronto, Ontario, Canada.

Manly, B.J.F., (1994) *Multivariate Statistical Methods - A primer* (2nd ed.). London: Chapman and Hall/CRC.

Minghim, R., & Forrest, A.R., (1995) An illustrated analysis of sonification for scientific visualization. *Proc. of IEEE Visualization 1995* (pp. 110-117). Atlanta, USA.

Ng, R., & Han, J., (1994) Efficient and effective clustering method for spatial data mining. *Proc. of the 20th VLDB Conference* (pp. 144–155). Santiago, Chile.

Pang. A.T., Wittenbrink, C.M., & Lodha, S.K, (1997) Approaches to uncertainty visualization. *The Visual Computer*, *13*(8), 370-390.

Santini, S., & Jain, R., (1999) Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(9), 871-883.

Silverman, B.W., (1986) *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall.

Telea, A., & van Wijk, J.J., (1999) Simplified Representation of Vector Fields. *Proc. of IEEE Visualization 1999* (pp. 35-42). San Francisco, USA.

Tilton, J.C., & Lawrence, W.T., (2000) Interactive Analysis of Hierarchical Image Segmentation. *Proc. of 2000 International Geoscience and Remote Sensing Symposium (IGARSS '00)* (Vol. 2, pp. 733-735). Honolulu, Hawaii, USA.

Wittenbrink, C.M., (1995) IFS fractal interpolation for 2D and 3D visualization. *Proc. of IEEE Visualization 1995* (pp. 77-85). Atlanta, USA.

Wittenbrink, C.M., Pang, A.T., & Lodha, S.K., (1996) Glyphs for visualizing uncertainty in vector fields. *IEEE Trans. on Visualization and Computer Graphics 2*(3), 266-279.