Interactive Visualization of Probability Density Functions using Clustering and Textures: A Case Study

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

[1] Department of Computer and Information Science, The Ohio State University, Columbus, Ohio, USA
[2] NASA Ames Research Center, Moffett Field, California, USA

Abstract

Visualizing and understanding probability density functions (pdf) on a two-dimensional domain is non-trivial because of their high dimensionality. Things become much more complicated when the data is in a three-dimensional space, and/or if it is time-varying. A pixel-by-pixel probing to visualize the pdfs can easily overwhelm the user. It is necessary to reduce the amount of information before rendering a visualization of the data. We present a clustering approach which presents a high level view by reducing the number of pdfs to visualize. We also discuss methods of texture synthesis to show average statistical measures of pdfs of the clusters.

1. Introduction

In many applications, uncertainties in the process of acquiring the data results in probabilistic data. In others, when only partial true (measured) data is available, simulations can be done to fill in missing data based on known statistical properties, resulting in data which is uncertain to some degree. The data can be two dimensional, as in the case of satellite imagery data. Or, it can exist in a three dimensional space, for example, when an experiment is run on a 3D region. In dynamic scenarios, the data depends on time as well, adding an extra dimension. The data at each grid point can be a probability density function, which defines the probabilities of the data values at that grid point. Or it can be the values for multiple realizations of the data, from which we can construct the probability density function. Henceforth, we will use the term *pdf* for probability density functions. Please note that distribution data sets are different from multivariate data sets in the sense that the values are for a single variable instead of multiple variables.

A dense global representation of the data, *i.e.*, drawing the *pdfs* at each data point is not trivial because of the high dimensionality of the data. If the user wants to visualize the actual *pdfs* at each grid point, he/she has to interactively probe each point, which can easily overwhelm the user. Our goal is to facilitate the users search by grouping together similar data and presenting representatives of such groups, thereby reducing the amount of data the user has to deal with. This aggregation of the data is implemented through clustering algorithms. The end result of the clustering is a sparse global representation of the data. The user can probe the representative *pdfs* of each cluster, and can interactively *merge* or *split* clusters to study the data at various levels of detail. Moreover, each cluster can be texture mapped with informative patterns, which convey various statistical summaries of the cluster, *e.g.*, mean, variance, skewness etc.

In the following section, we discuss the related work in the fields of uncertainty visualization and clustering (§ 2). In § 3, we present the datasets and briefly go over the stages of data measurement and statistical data generation. The clustering algorithm for different types of probability data is discussed in § 4, and visualization of clustering results using textures is explained in § 5.

2. Related Work

The problem of uncertainty visualization has inspired a broad variety of solutions ranging from uncertainty glyphs [14, 19] to using sound cues [11, 13], and from procedural annotations [2] to geometric effects [1, 14, 18, 12]. These methods do not, however, permit the visualization of the exact *pdf*. Some of these methods consider uncertainty to be a scalar quantity for the purposes of the visualization technique. The visualization of the data then boils down to showing two values for each data point: the expected value of the data (*e.g.*, mean) and the error in the data, which might be approximated by, say, variance. Not visualizing the entire *pdf* deprives the user of additional, possibly interesting, information. Moreover, this has the disadvantage of possible misinterpretation of the data. For example, in the extreme case, the mean of a variable with a bi-modal *pdf* can lie in between the two peaks; but the probability of the variable getting a value anywhere near the mean might be zero. Ehlschlaeger *et.al.*[5] use animations of multiple realizations of spatial data to help the user gain insight to the uncertainty in the realizations. Techniques have been developed to study the *pdfs* in greater detail by Kao *et.al.*[8]. They use various statistical summaries of the *pdfs* (*e.g.*, mean, variance, skewness, kurtosis, inter-quartile distance) to construct a dense global visualization of dataset. However, the user has to probe each point to visualize the actual *pdf*. In [9], Kao *et.al.* present additional methods of visualizing 2D *pdf* data using cutting planes, surface graphs, *pdf* isosurfaces etc. Our hierarchical clustering gives a multi-resolution representation of the *pdf* data, and the user can interactively visualize the *pdfs* for each cluster at various levels of detail.

Clustering is a very powerful tool, and has been used in various flavors in almost every scientific community. The whole literature is too wide to mention here, so we will mention some of the research that is directly related to our work. Some of the recent clustering algorithms are K-means, Pam, Clarans, DBScan, Cure, Rock and Chameleon. Any clustering algorithm can be used for our purposes; however hierarchical clustering is better as it allows the user to view the clustering in different levels of detail [7, 16]. We use an agglomerative (bottom-up) clustering similar to James Tilton's recursive hierarchical image segmentation for satellite spectral data which alternates between region growing and spectral clustering of similar regions at every step[17]. The algorithm is very computationally expensive as it calculates the distance (Euclidean Spectral distance) between every pair of clusters at each step. In our algorithm, we only calculate the distance between neighboring pairs of clusters.

3. Application Data Sets

We present our *pdf* visualization techniques with two conditional simulation datasets. The first data set is from data constructed using a small region in the Netherlands imaged by the Landsat Thematic Mapper [4]. For this dataset, the bio-physical variable to be 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 distribution data set was generated using this information: conditional co-simulation[3] using both ground measurements and the coincident satellite image. The data set consists of 101 × 101 pixels and 250 realizations. Values range from 0 to 255, re-scaled from percentage cover [9].

Our second distribution data set 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 such as temperature and salinity. To dynamically evolve the physical uncertainty, an Error Subspace Statistical Estimation (ESSE) scheme [10] 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. This gives us the raw data for a 3D distribution data set. The field value is for sound speed and is derived from the other physical field values. The dimension of this dataset is 65 × 72 × 42, with 80 realizations of the volume.

4. Clustering framework

We use a hierarchical clustering framework as a basis for the visualization techniques presented in this paper. A hierarchical clustering (as opposed to a non-hierarchical one) of the data allows the scientist to interact with it, and study the dataset at different levels of detail. Such interactivity goes a long way towards a faster and better appreciation of the data, specially if the data is high-dimensional, and if it is a large dataset. The higher levels of the cluster tree provide a global view to the interested scientist; and the lower levels provide details when the region of interest is smaller (figure 1). We use a bottom-up clustering method for our implementation. The visualization methods presented in this paper are not dependent on which clustering method is used, and any hierarchical scheme will serve our purpose.

In the rest of this section, we present the clustering algorithm, and the distance functions used. We also discuss the clustering for time-varying data.

4.1. Distance Functions

Distribution datasets may come in the form of *pdfs*, or they may be in the form of multiple realizations from random experiments. For example, both the datasets mentioned in § 3 are generated by conditional simulations, which result in many realizations of the data. If we think of each data point (each point on the rectilinear grid) in the dataset as one random variable, then each realization can be considered as the result of an experiment where the random variable got the

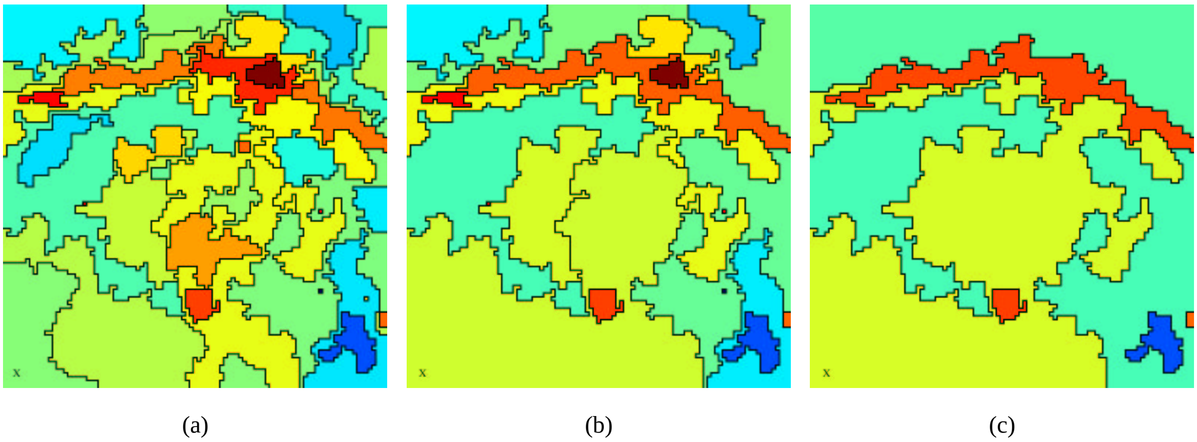


Figure 1: *Hierarchical clustering of the Landsat distribution data for different error thresholds: (a) low error, (b) medium error, and (c) high error threshold. The clusters are colored by their mean values. The colormap used gives blue for low values, green for medium, and red for high values.*

value given by the realization. One solution for clustering multiple realization data is to convert the data form to *pdfs*, and run the clustering algorithm as if the data handed to us was *pdf* data. A histogram is a crude way of approximating the *pdf*. More accurate ways to construct the *pdf* are the naive estimator and the kernel estimator[9, 15]. Many alternatives are available for use as the distance function for *pdf* clustering: Kullback-Leibler (KL) distance, entropy, manhattan or Euclidean distance between two histogram vectors, the dot-product between two histogram vectors etc.

However, we loose spatial (spatio-temporal, for time-varying data) correlation information while converting multiple realizations of the data to *pdfs*. Clustering the realization data itself preserves the spatial information. For example, two grid points which have similar *pdfs* may have opposing behaviors. Consider a random variable X that has a *pdf* which is symmetric about some value; one can then construct a new random variable Y with exactly the same *pdf*, such that X and Y have a correlation of -1.0. It would be counter-intuitive to cluster X and Y together. Hence, for our two datasets, we chose not to do a *pdf* estimation before clustering. Instead, we stack all the realizations for each grid point to form a *realization vector*. We then define the distance between two data points as the Manhattan (taxicab) distance between their realization vectors. Alternatively, other distances like the Euclidean distance can be used.

4.2. Probability Data Clustering

We use a bottom-up clustering technique to create a tree of clusters, also known as *dendrogram*. At first, each grid point is defined to be a cluster by itself. We then find and merge the two neighboring clusters whose inter-cluster distance is the minimum. We define the *inter-cluster distance* between cluster A and cluster B as the maximum distance between any data point in A and any data point in B. The new cluster (i.e., the parent cluster) of A and B is then associated with an *intra-cluster error*, which is defined as the maximum of the following three values: intra-cluster error of A, intra-cluster error of B, or the inter-cluster distance between A and B. The clustering process is continued iteratively till we are left with a single cluster that contains all the points in the 2-D domain. This last cluster is the root of the dendrogram, which is the end result of the clustering algorithm. To restrict the size of the cluster-tree, we maintain only those clusters in the tree whose cardinality is greater than a threshold *MinClusterSize* given during the clustering process. In general, every node in the tree (except the leaf nodes) will have two children. Further reduction of the tree size is possible by throwing away alternate levels of the tree; each node will then have four children. Note that other inter-cluster distances, e.g., distance between the centroids, hausdorff distance, etc. can be used.

The user can visualize the clustering results at various levels of detail by changing a threshold value for the intra-cluster error (see figure 1). The cluster-tree is traversed recursively and only clusters whose intra-cluster error is less than the given threshold are shown. Higher thresholds will result in fewer and bigger clusters compared to lower thresholds. The user can also interactively visualize the clustering results by manually going through the dendrogram using a GUI. He/she can click on a node of the dendrogram to split/unsplit the cluster corresponding to that particular node. To make the visual exploration process less taxing for the user, it is desirable that the color of the children (of the cluster being split) have some resemblance to color of the parent. We use the mean of a cluster to determine its color, as the means of the children are closer to each other and to the parent than an unrelated node. One might argue that the visual impact is weakened by rendering two neighboring clusters with possibly very similar colors. In § 5, we propose the use of textures which can provide visual cues to more statistical summaries of the clusters, and thus highlight the differences between the two neighbors.

The statistical properties of a cluster can be approximated by using the realizations of all the data points within that cluster. An 'average' *pdf* can be constructed in this manner. Figure 2 shows the 'average' *pdf* for the cluster containing the lower left corner of the Landsat dataset. Depending on the level of the cluster in the dendrogram, i.e., towards the root or near the leaves, the average *pdf* will be respectively less or more close to the actual *pdfs* of the points inside the cluster. In figure 2(a), which corresponds to a low error threshold, the cluster 'average' *pdf* is very close to the actual *pdf* of the data point 'X' at the lower left corner. The red curve shows the variance of each bin of the actual *pdfs* of the data points within the cluster.

4.3. Time-series Probability Data Clustering

In this section, we discuss the case when the distribution data is a time dependent. At the time of writing this case study, we do not have access to time-dependent distribution data. We use our 3D (ocean) dataset as an example of time-varying data. The z-axis, *i.e.*, the ocean depth, is assumed to be the time dimension. We will consider two methods of clustering and visualizing a time-varying dataset.

4.3.1. Spatial Clustering

Spatial clustering is useful when the user is interested in the temporal behavior of fixed regions in the spatial domain. For example, in case of geographical datasets, the user might be interested in the temporal behavior of particular regions. In this case, the time dependent behavior of the data points is the attribute used as a (dis)similarity measure. The distance between any two grid points is defined as the distance between their time-series. For our implementation, we take the distance between two time-series as the sum of the distances between the two at each time-step. In other words, the distance is defined as the Manhattan distance between the two time-series. As in § 4.2, the distance between the two for each time-step is the Manhattan distance between the two realization vectors at that time-step. Like the previous section, the clustering results can be visualized at various levels of detail. For each cluster, the user can then explore how the *pdf* changes with time.

4.3.2. Spatio-temporal Clustering

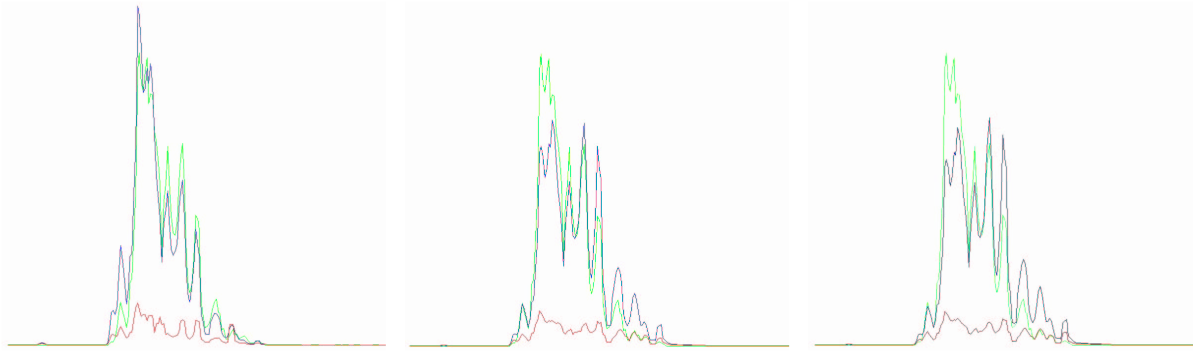
Spatio-temporal information can often turn out to be more useful to the user than visualizing spatial and temporal information separately. Our clustering framework can be extended to take into account both the spatial and temporal dimensions simultaneously. The clustering algorithm remains the same, but now it is run assuming the data to be three dimensional, i.e., we consider time to be another dimension in space-time. The data at each point in this case is a realization vector, and the clustering is essentially a 3D extension of § 4.2.

The results can be visualized in various levels of detail by selecting a threshold value for the intra-cluster error. We can visualize the 3D clustering results as an animation of 2D slices of the 3D volume, where each slice is perpendicular to the time-axis. We start by showing the clusters on the first slice (slice which intersects the time-axis at the initial time-step), and then play an animation by moving the slice along the time-axis. Each frame of the animation now shows the clustering results on the 2-D spatial domain for a given time-step. Since the correspondence between clusters on neighboring slices is known (corresponding clusters come from the same 3D cluster), they have the same color or texture. This allows the user to visually track the movement of the clusters. This method of clustering can be a very useful tool for meteorological and EOS data. Scientists can visualize the movement of clusters on the earth's surface. For example, if the data is time-series of surface temperature, then temperature clusters in the northern hemisphere should move southwards as winter approaches. Figures 3 and 4 show results of a 3D clustering on the ocean data.

5. Cluster Visualization

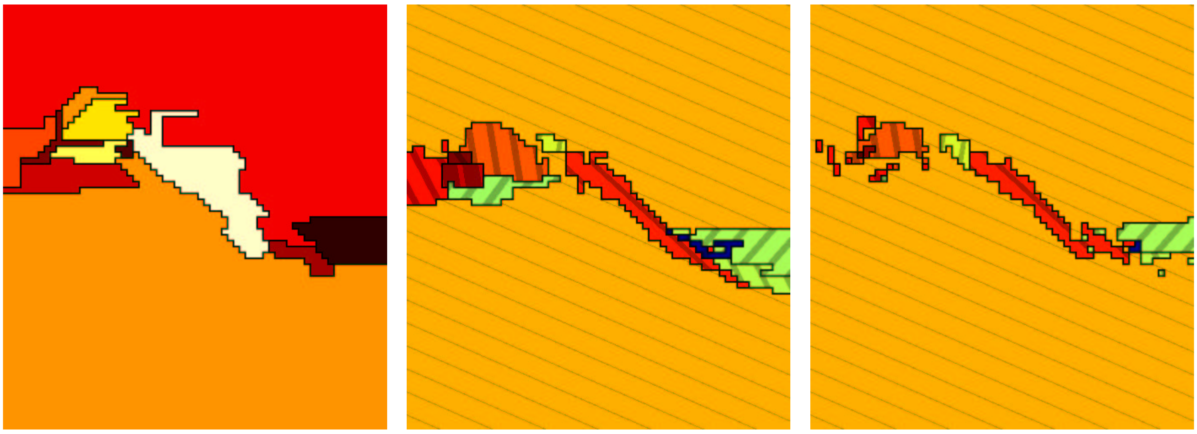
In figure 1, we use the means of clusters to determine their colors. In addition to giving similar colors to siblings and their parents in the hierarchical tree (§ 4.2), it also gives the user some information about the cluster. Next, we discuss ways to show more statistical information using patterns on the clusters.

Along with mean and variance, other statistical summaries such as the skewness, kurtosis, and the inter-quartile range can be very helpful in understanding the *pdf*. 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*. For figures 3 and 4, we use variance and skewness as two extra inputs while rendering the clusters. The variance and skewness are remapped to a range of [-1,1] for convenience. The uniform mean color is now replaced with alternating lighter and darker bands. 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. For the third statistical variable (skewness), 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 rotation (for values of -1 or 1) is a little less than ninety degrees. Looking at figures 3(b,c) and 4, we can see that most of clusters have



(a) (b) (c)

Figure 2: *Probability Density function for the point 'X' on the lower left corner of the Landsat dataset is shown in green. Figures (a),(b), and (c) respectively show the average pdf (blue) for the clusters containing that point in figure 1(a),(b) and (c). The 'average' pdf of the cluster is quite close to the pdf of 'X' in figure (a), but deviates in figures (b) and (c). The red graph shows the variance for each bin of the actual pdfs of all the data points within the cluster.*



(a) (b) (c)

Figure 3: *Different clustering results for the ocean dataset: (a) Spatial clustering of the ocean dataset, assuming time to be along z axis, and 3D clustering of the ocean dataset:(b) clusters at the surface of the ocean (z=1), and in the middle (c) z = 21.*

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. It is also apparent that, all the clusters other than the large brown cluster have high variance.

6. Conclusion

We have implemented a hierarchical clustering scheme for distribution data which allows for a multiple level of detail exploration of dataset. Combined with other interactive methods, this can prove to be very useful for scientists in studying probability density data.

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 Alex Pang and Alison Luo for very helpful discussions on the subject.

References

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

2. A. Cedilnik and P. Rheingans. Procedural annotation of

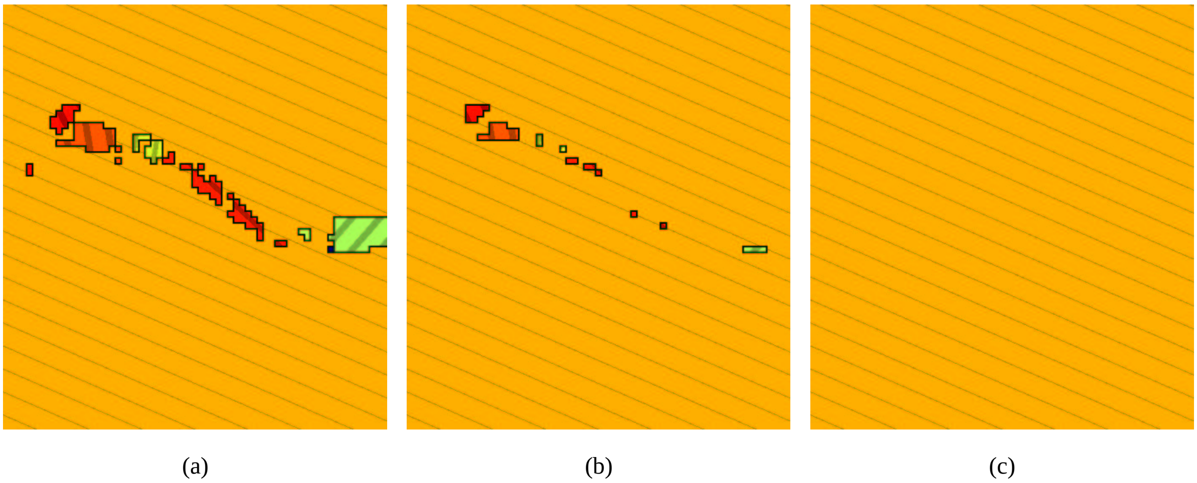


Figure 4: *3D clustering of the ocean dataset: (a) Clusters at level z = 25, (b)z = 29, and (c) z = 34. The clusters from upper layers of the dataset start breaking and finally vanish. In case of time-varying data, cluster movement can be visually tracked.*

uncertain information. *Proceedings of IEEE Visualization '00*, 77-84, 2000.

3. C.V. Deutsch and A. G. Journel. *GSLIB: Geostatistical Software Library*. Oxford University Press, New York, 1998.
4. J. L. Dungan. Conditional simulation: An alternative to estimation for achieving mapping objectives. F. van der Meer A. Stein and B. Gorte, editors, *Spatial Statistics for Remote Sensing*, 135-152, Kluwer, Dordrecht, 1999.
5. C.R. Ehlschlaeger, A.M. Shortridge, and M.F. Goodchild. Visualizing spatial data uncertainty using animation. *Computers in GeoSciences*, **23**(4):387-395, 1997.
6. G. Grigoryan and P. Rheingans. Probabilistic Surfaces: Point Based Primitives to Show Surface Uncertainty. *Proceedings of IEEE Visualization '02*, 147-154, 2002.
7. B. Heckel, G. Weber, B. Hamann, and K. Joy. Construction of Vector Field Hierarchies. *Proceedings of IEEE Visualization '99*, 19-27, 1999.
8. D. Kao, J. Dungan, and A. Pang. Visualizing 2D Probability Distributions from EOS Satellite Image-Derived Data Sets: A Case Study. *Proceedings of IEEE Visualization '01*, 457-460, 2001.
9. D. Kao, A. Luo, J. L. Dungan, and A. Pang. Visualizing Spatially Varying Distribution Data. *Proceedings of Information Visualization '02*, 219-225, 2002.
10. P.F.J. Lermusiaux. Data assimilation via error subspace statistical estimation, Part II: Middle Atlantic Bight shelfbreak front simulations and ESSE validation. *Monthly Weather Review*, **127**(7):1408-1432, 1999.
11. S. K. Lodha, C. M. Wilson, and R. E. Sheehan. LISTEN: sounding uncertainty visualization. *Proceedings of IEEE Visualization '96*, 189-195, 1996.
12. S. Lodha, R. Sheehan, A. Pang, and C. Wittenbrink. Visualizing geometric uncertainty of surface interpolants. *Proceedings of Graphics Interface*, 238-245, 1996.
13. R. Minghim and A.R. Forrest. An illustrated analysis of sonification for scientific visualization. *Proceedings of IEEE Visualization '95*, 110-117, 1995.
14. A. T. Pang, C. M. Wittenbrink, and S. K. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, **13**(8):370-390, 1997.
15. B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall, 1986.
16. A. Telea and J.J. van Wijk. Simplified Representation of Vector Fields. *Proceedings of IEEE Visualization '99*, 35-42, 1999.
17. J.C. Tilton and W.T. Lawrence. Interactive Analysis of Hierarchical Image Segmentation. *Proceedings of the 2000 International Geoscience and Remote Sensing Symposium (IGARSS '00)*, 2000.
18. C. M. Wittenbrink. IFS fractal interpolation for 2D and 3D visualization. *Proceedings of IEEE Visualization '95*, 77-84, 1995.
19. C. M. Wittenbrink, A. T. Pang, and S.K. Lodha. Glyphs for visualizing uncertainty in vector fields. *IEEE Transactions on Visualization and Computer Graphics*, **2**(3):226-279, 1996.