# A Study of Hierarchical Correlation Clustering for Scientific Volume Data

Yi Gu and Chaoli Wang

Michigan Technological University

**Abstract.** Correlation study is at the heart of time-varying multivariate volume data analysis and visualization. In this paper, we study hierarchical clustering of volumetric samples based on the similarity of their correlation relation. Samples are selected from a time-varying multivariate climate data set according to knowledge provided by the domain experts. We present three different hierarchical clustering methods based on quality threshold, k-means, and random walks, to investigate the correlation relation with varying levels of detail. In conjunction with qualitative clustering results integrated with volume rendering, we leverage parallel coordinates to show quantitative correlation information for a complete visualization. We also evaluate the three hierarchical clustering methods in terms of quality and performance.

## 1 Introduction

Finding connection among time-varying multivariate data is critically important in many areas of scientific study. In the field of visualization, researchers have investigated relationships among variables and developed techniques to visualize them. One effective solution is to cluster voxels based on correlation similarity. This allows users to observe how those voxels that have similar correlation behaviors distribute over space and detect possible patterns. When the size of volume data is large, we can select samples for computation to gain an overall impression of the correlation relation in a cost-effective manner. Many research efforts adopted the standard correlation coefficients to study the linear correlation between variables, yet little work is done to build a hierarchy for coarse-to-fine exploration of data correlation. Hierarchical clustering can show cluster within clusters and much as in multiresolution visualization, it provides us a flexible means to adaptively examine the data. In this paper, we present three different hierarchical clustering methods for correlation classification and perform a comparative study of their quality and performance using a climate data set. The evaluation includes side-by-side qualitative comparison of clustering results and quantitative comparison using silhouette plot. We conclude this paper by making our recommendation and pointing out our future research.

## 2 Related Work

Analyzing and visualizing time-varying multivariate data remains a significant challenge in visualization research. Over the years, researchers have applied the

standard pointwise correlation in their analysis [1–4]. New user interfaces were also developed to visualize multivariate data relationships [1, 2]. To the best of our knowledge, hierarchical clustering of time-varying volume data based on the similarity of multivariate correlation has not been investigated, which is the focus of this work.

Parallel coordinates have become a popular technique for visualizing relationships among a large collection of variables. An important issue for parallel coordinates is to order dimensions to reveal multivariate data patterns. One way to achieve this is based on the evaluation of similarity between dimensions. Ankerst et al. [5] provided global and partial similarity measures for two dimensions. They defined the dimension arrangement problem as an optimization problem to minimize the summation of the dissimilarity of all consecutive pairs of axes. Yang et al. [6] built a hierarchical dimension structure and allowed dimension reordering and filtering. We utilize parallel coordinates to show quantitative correlation information for volume samples that are clustered hierarchically. To effectively present relationships among samples, we define two correlation-based distance measures for dimension clustering and ordering.

## 3 Sample Selection

Given a large time-varying multivariate data set, computing the correlation among all voxels over all time steps could be very expensive. A viable alternative is to sample in space and time. This is feasible because in general, the correlation pattern with respect to close neighboring reference locations are similar and not all time steps are necessary in order to detect the correlation pattern. Thus, we can compute the correlation for selected samples at selected time steps and perform clustering to gain an overview of the correlation relationships.

We can adopt uniform or random sampling depending on the need. The domain knowledge about the data can also help us choose a customized sampling scheme. For the climate data set we experiment with, the domain scientists provide the knowledge to assist us in choosing spatial samples and time steps.

## 4 Hierarchical Correlation Clustering

### 4.1 Correlation Matrix

We use the Pearson product-moment correlation coefficient to evaluate the linear correlation between the time series at two sampling locations $X$ and $Y$

$$\rho_{XY} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{X_t - \mu_X}{\sigma_X}\right)\left(\frac{Y_t - \mu_Y}{\sigma_Y}\right), \tag{1}$$

where $T$ is the number of time steps. $\mu_X$ ($\mu_Y$) and $\sigma_X$ ($\sigma_Y$) are the mean and standard deviation of $X$ ($Y$), respectively. $\rho_{XY}$ is in $[-1, 1]$. The value of 1 (-1) means that there is a perfect positive (negative) linear relationship between $X$

and $Y$. The value of 0 shows that there is no linear relationship between $X$ and $Y$. For all the samples given, we can build a correlation matrix $\mathbf{M}$ with $\mathbf{M}_{i,j}$ recording $\rho_{X_i X_j}$. If $X_i$ and $X_j$ are drawn from the same variable (two different variables), then $\mathbf{M}$ is the self-correlation (cross-correlation) matrix.

### 4.2 Distance Measure

Before clustering the samples, we need to define the distance between two samples $X$ and $Y$. In this paper, we take two different distance measures which both take the correlation matrix $\mathbf{M}$ as the input. The first distance measure only considers $\mathbf{M}_{i,j}$ for samples $X_i$ and $X_j$, and we define the distance as

$$d_s(X_i, X_j) = 1 - |\mathbf{M}_{i,j}|. \tag{2}$$

That is, the distance indicates the strength of linear correlation between $X_i$ and $X_j$. When $X_i$ and $X_j$ are perfectly correlated (regardless of the sign), then $d_s(X_i, X_j)$ gets its minimum of 0. If $X_i$ and $X_j$ have no linear correlation, then $d_s(X_i, X_j)$ gets its maximum of 1. The second distance measure considers two rows $\mathbf{M}_{i,k}$ and $\mathbf{M}_{j,k}$ for samples $X_i$ and $X_j$, and we define the distance as

$$d_v(X_i, X_j) = \sqrt{\sum_{k=1}^{N} \left(\mathbf{M}_{i,k} - \mathbf{M}_{j,k}\right)^2}, \tag{3}$$

where $N$ is the number of samples. We compute $d_v(X_i, X_j)$ for all pairs of samples and normalize them to $[0, 1]$ for our use.

### 4.3 Hierarchical Clustering

The correlation matrix and distance measure defined above can be used to cluster the samples in a hierarchical manner. In general, there are two approaches to build such a hierarchy, *agglomerative* or *divisive* [7]. The agglomerative (or "bottom-up") approach starts with each sample in its own cluster and merges two or more clusters successively until a single cluster is produced. The divisive (or "top-down") approach starts with all samples in a single cluster and splits the cluster into two or more clusters until certain stopping criteria are met or each sample is in its own cluster. The advantages of hierarchical clustering are that it can show "cluster within clusters" and it allows the user to observe clusters according to the depth-first-search or breadth-first-search traversal order. We refer interested readers to the work of Zimek [8] for the mathematical background of correlation clustering. In this paper, we experiment with three hierarchical clustering methods based on quality threshold, k-means, and random walks to investigate the correlation relation among samples at different levels of detail.

**Hierarchical Quality Threshold** This is a bottom-up hierarchical clustering approach which uses a list of distance thresholds $\{\delta_0, \delta_1, \delta_2, \ldots, \delta_l\}$ to create a hierarchy of at most $l+1$ levels in $l$ iterations. These thresholds must satisfy the following conditions: $\delta_i < \delta_j$ if $i < j$; $\delta_i \in (0, 1)$ for $1 < i < l-1$; and $\delta_0 = 0$, $\delta_l = 1$. At the beginning, each sample is in its own cluster. At the first iteration, we build a candidate cluster for each sample $s$ by including all samples that have their distance to $s$ smaller than threshold $\delta_1$. Then, we save the cluster with the largest number of samples as the first true cluster and remove all samples in this cluster from further consideration. In the true cluster, sample $s$ is treated as its representative sample. We repeat with the reduced set of samples until all samples are classified. At the second iteration, we use threshold $\delta_2$ to create the next level of hierarchy. The input to this iteration is all representative samples gathered from the previous iteration. We continue this process for the following iterations until we finish the $l$th iteration or until we only have one cluster left in the current iteration.

**Hierarchical k-Means** The popular k-means algorithm classifies $N$ points into $k$ clusters, $k < N$. Generally speaking, the algorithm attempts to find the natural centers of $k$ clusters. In our case, the input is the $N \times N$ correlation matrix $\mathbf{M}$ where each row in $\mathbf{M}$ represents a $N$-dimensional sample to be classified. The k-means algorithm randomly partitions the input points into $k$ initial clusters and chooses a point from each cluster as its centroid. Then, we reassign every point to its closest centroid to form new clusters. The centroids are recalculated for the new clusters. The algorithm repeats until some convergence condition is met. We extend this k-means algorithm for the top-down hierarchical clustering where we take each cluster output from the previous iteration as the input to the k-means algorithm and construct the hierarchy accordingly. This process continues until a given number of levels is built or the average distortion within every cluster is less than the given threshold.

**Random Walks** Random walks [9] are also a bottom-up hierarchical clustering algorithm. In our case, this algorithm considers the $N \times N$ correlation matrix $\mathbf{M}$ as a fully connected graph where we treat $|\mathbf{M}_{ij}|$ as the weight for edge $e_{ij}$. At each step, a walker starts from vertex $v_i$ and chooses one of its adjacent vertices to walk to. The probability that an adjacent vertex $v_j$ is chosen is defined as $\mathbf{P}_{ij} = |\mathbf{M}_{ij}|/d_i$, where $d_i = \sum_{j=1}^{N} |\mathbf{M}_{ij}|$. In this way, we can compute a random walk probability matrix $\mathbf{P}^t$ to record the possibility starting from $v_i$ to $v_j$ in $t$ steps. With $\mathbf{P}^t$, we define the distance between $v_i$ and $v_j$ as

$$r_{ij}^t = \sqrt{\sum_{k=1}^{N} \frac{\left(\mathbf{P}_{ik}^t - \mathbf{P}_{jk}^t\right)^2}{d_k}}, \tag{4}$$

Random walks start with every vertex in its own cluster. Then the algorithm iteratively merges two clusters with the minimum mean distance into a new

cluster, and updates all the distances between clusters. This process continues until we only have one single cluster left. The probability of going from a cluster $C$ to $v_j$ in $t$ steps is defined as

$$\mathbf{P}_{Cj}^t = \frac{1}{|C|} \sum_{i \in C} \mathbf{P}_{ij}^t, \tag{5}$$

and the distance between two clusters $C$ and $D$ is defined as

$$r_{CD}^t = \sqrt{\sum_{k=1}^{N} \frac{\left(\mathbf{P}_{Ck}^t - \mathbf{P}_{Dk}^t\right)^2}{d_k}}. \tag{6}$$

## 5 Evaluation

To evaluate the effectiveness of different hierarchical clustering algorithms, we generate the same or very similar number of clusters for all methods for a fair comparison. A straightforward comparison is to directly compare the clustering results side by side in the volume space. The limitation of this comparison is that it is subjective, which can be complemented by a quantitative comparison.

Silhouette plot [10] is a technique to verify the quality of a clustering algorithm and it works as follows. For each point $p_i$ in its cluster $C$, we calculate $p_i$'s average similarity $a_i$ with all other points in $C$. Then for any cluster $C_j$ other than $C$, we calculate $p_i$'s average similarity $d_{ij}$ with all points in $C_j$. Let $b_i$ be the minimum of all $d_{ij}$ for $p_i$, and the corresponding cluster be $C_k$ (i.e., $C_k$ is the second best cluster for $p_i$), we define the silhouette value for $p_i$ as

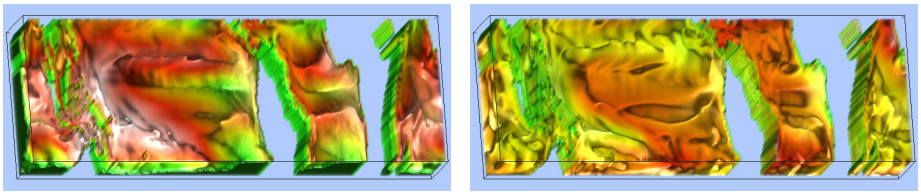
$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}. \tag{7}$$

$s_i$ is in the range of $[-1, 1]$. If $s_i$ is close to 1 (-1), it means $p_i$ is well (poorly) clustered. If $s_i$ is near 0, it means $p_i$ could be in either cluster. If $\max(s_i) < 0.25$ for all the points, it indicates that these points are poorly clustered. There are two possible reasons. One reason is that the points themselves could not be well separated or clustered. Another reason is that the clustering algorithm does not perform well. To draw the silhouette plot, we sort $s_i$ for all the points in each cluster and display a line segment for each point to show its silhouette value. By comparing the silhouette plots for all three clustering algorithms, we can evaluate their effectiveness in a quantitative manner.

## 6 Results and Discussion

### 6.1 Data Set

We conducted our hierarchical correlation clustering study using the tropical oceanic data simulated with the National Oceanic and Atmospheric Administration (NOAA) Geophysical Fluid Dynamics Laboratory (GFDL) CM2.1 global



(a) temperature field (b) salinity field

**Fig. 1.** Snapshots of the temperature and salinity fields at the first time step. Green, yellow, and red are for low, medium, and high scalar values, respectively.

coupled general circulation model. The equatorial upper-ocean climate data set covers a period of 100 years, which is sufficient for our correlation study. The data represent monthly averages and there are 1,200 time steps in total. The spatial dimension of the data set is $360 \times 66 \times 27$, with the $x$ axis for longitude (covering the entire range), the $y$ axis for latitude (from 20°S to 20°N), and the $z$ axis for depth (from 0 to 300 meters). Figure 1 shows the two fields, temperature and salinity, which we used in our experiment. The results we reported are based on the temperature and salinity cross correlation.

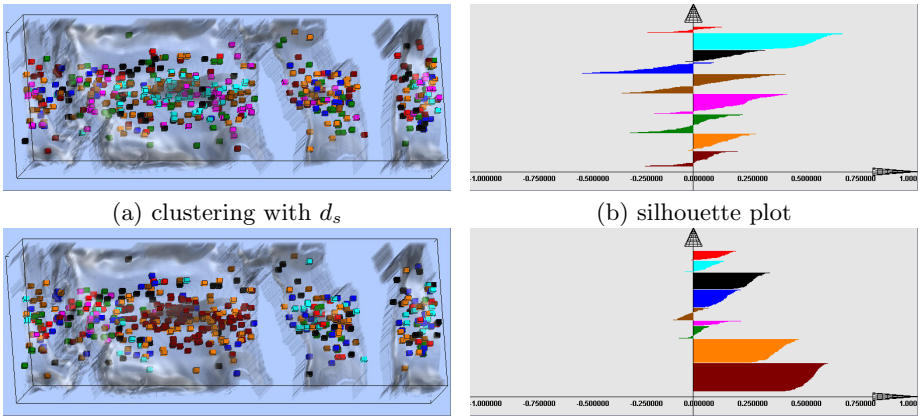
### 6.2 Sampling in Space and Time

For this climate data set, the NOAA scientists provided us with the following knowledge for sample selection. First, voxels belong to the continents are not considered. Second, voxels near the Earth's equator are more important than voxels farther away. As such, the simulation grid along the latitude is actually non-uniform: it is denser near the equator than farther away. Third, voxels near the sea surface are more important than voxels farther away. We incorporated such knowledge into sample selection. Specifically, we used a Gaussian function for the latitude (the $y$ axis) and an exponential function for the depth (the $z$ axis) to compute the probability of a voxel being selected. This treatment allows us to sample more voxels from important regions. It also agrees well with the computational grid used in simulation. In our experiment, we sampled two sets of voxels (500 and 2000) from the volume for correlation clustering.

We only took a subset of time steps from the original time series to reduce the computation cost in the correlation study. As suggested by the scientists, we strode in time to reduce the data volumes with fairly independent samples: we took the first time step, then chose every 12th time step (i.e., we picked the volumes corresponding to the same month). A total of 100 time steps were selected to compute the correlation matrix.

### 6.3 Distance Measure Comparison

In Figure 2, we show the comparison of two distance measures $d_s$ and $d_v$ on the clustering performance while all other inputs are the same. Although it is



(a) clustering with $d_s$ (b) silhouette plot

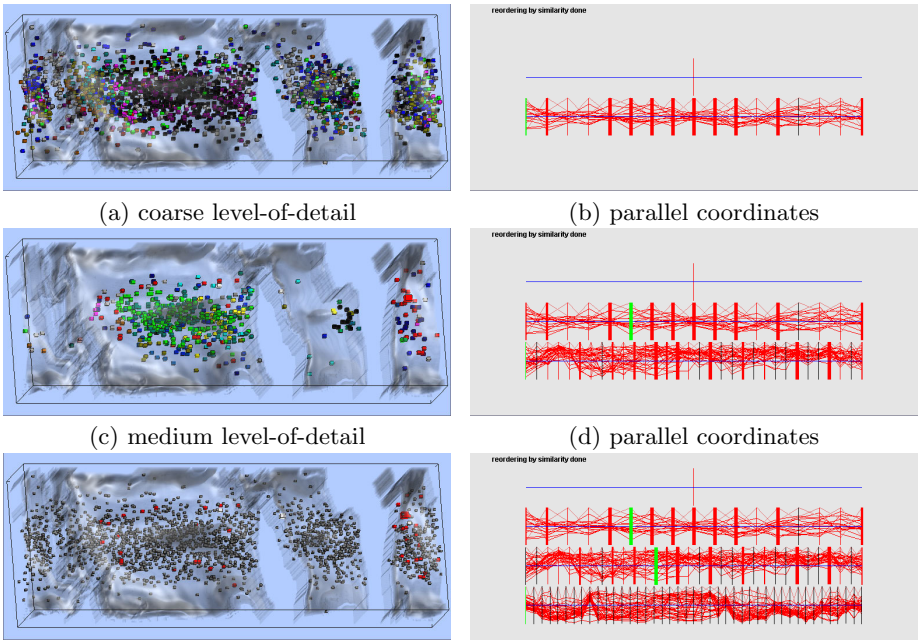
(c) clustering with $d_v$ (d) silhouette plot

**Fig. 2.** Comparison of two distance measures $d_s$ and $d_v$ with random walks. Both have 500 samples and produce nine clusters which are highlighted with different colors in (a) and (c). From (b) and (d), we can see that $d_v$ performs better than $d_s$.

not obvious from the clustering results, the silhouette plots shown in (b) and (d) clearly indicate that $d_v$ is better than $d_s$. In this example, 45.4% of samples have their silhouette value larger than 0.25 when using $d_v$, compared with 22.4% of samples using $d_s$. For samples with silhouette value less than 0.0, it is 8.8% with $d_v$ and 26.8% with $d_s$. The reason that $d_v$ performs better is because given two samples, $d_v$ considers correlations between all samples while $d_s$ only considers the correlation between the two samples. The same conclusion can be drawn for the other two hierarchical clustering algorithms. We thus used $d_v$ as the distance measure in all the following test cases.

### 6.4 Level-of-Detail Correlation Exploration

Figure 3 shows the level-of-detail exploration of correlation clusters with the hierarchical quality threshold algorithm. Samples that are not in the current level being explored can be either hidden or de-emphasized as shown in (c) and (e), respectively. Parallel coordinates show the correlation relation quantitatively. In our case, the number of axes in a level equals the number of samples. The thickness of each axis is in proportion to the number of samples it contains in the next level, which provides hint for user interaction. The user can simply click on an axis to see the detail or double click to return. For each level in the parallel coordinates, we sort the axes by their similarity so that sample correlation patterns can be better perceived. The samples along the path from the root to the current level are highlighted in white and green in the volume and parallel coordinates views, respectively. By linking the parallel coordinates view with the volume view, we enable the user to explore the hierarchical clustering results in a controllable and coordinated fashion.




(a) coarse level-of-detail (b) parallel coordinates

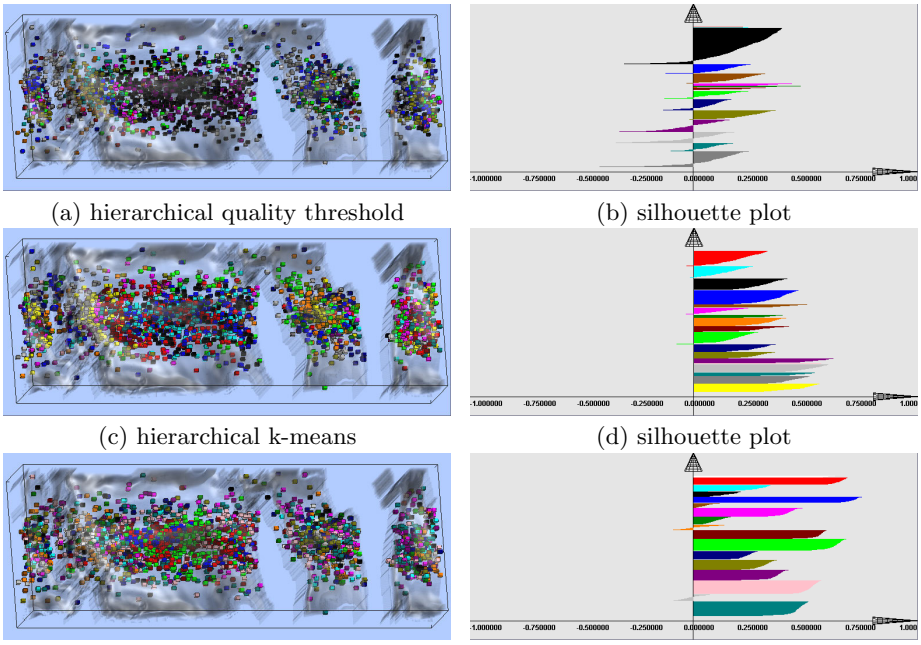
(c) medium level-of-detail (d) parallel coordinates

(e) fine level-of-detail (f) parallel coordinates

**Fig. 3.** Level-of-detail exploration of correlation clustering of 2000 samples with hierarchical quality threshold. (a) shows all the samples, (c) shows only samples in the current level, and (e) de-emphasizes samples that are not in the current level using gray color and smaller size. Parallel coordinates show the qualitative correlation relationships among samples accordingly. The axes in the current level are reordered by their similarity with each axis corresponding to a (representative) sample.

### 6.5 Clustering Algorithm Comparison

In Figure 4 and Table 1, we compare the three hierarchical clustering algorithms. Their silhouette plots clearly indicate that hierarchical quality threshold performs the worst. While hierarchical k-means and random walks have comparable performances in terms of percentages of samples with silhouette value larger than 0.25 and smaller than 0.0. Random walks have more samples with silhouette value larger than 0.5 and it take much less time to compute compared with hierarchical k-means. Therefore, the random walks algorithm is the best in terms of quality and performance tradeoff. Unlike quality threshold and k-means algorithms, random walks do not require parameters such as the threshold or number of clusters to start with, which also makes it appealing for use. On the other hand, we observed that the timing of quality threshold is very sensitive to the number of levels and the threshold chosen for each level. The output and resulting quality are also very unstable. The quality of k-means is fairly good except that it requires much more time to compute.



(a) hierarchical quality threshold (b) silhouette plot

(c) hierarchical k-means (d) silhouette plot

(e) random walks (f) silhouette plot

**Fig. 4.** Comparison of the three hierarchical clustering algorithms with 2000 samples. The numbers of clusters generated are 17, 18, and 17 for quality threshold, k-means, and random walks respectively. From (b), (d), and (f), we can see that random walks produce the best result while quality threshold produces the worst result.

## 7 Conclusions and Future Work

We have presented a study of hierarchical correlation clustering for time-varying multivariate data sets. Samples are selected from a climate data set based on domain knowledge. Our approach utilizes parallel coordinates to show the quantitative correlation information and silhouette plots to evaluate the effectiveness of clustering results. We compare three popular hierarchical clustering algorithms in terms of quality and performance and make our recommendation. In the future, we will evaluate our approach and results with the domain scientists. We also plan to investigate the uncertainty or error introduced in our sampling in terms of clustering accuracy.

## Acknowledgements

This work was supported by Michigan Technological University startup fund and the National Science Foundation through grant OCI-0905008. We thank Andrew T. Wittenberg at NOAA for providing the climate data set. We also thank the anonymous reviewers for their helpful comments.

| | quality threshold | k-means | random walks |
|---|---|---|---|
| strategy | agglomerative | divisive | agglomerative |
| parameters | # levels<br>threshold for each level | # initial clusters<br>termination threshold | none |
| randomness | no | yes | yes |
| tree style | general | general | binary |
| speed | unstable (184.4s, 22.2s) | slow (673.9s, 30.1s) | fast (188.4s, 4.5s) |
| quality | bad (22.0%, 67.6%) | good (65.1%, 60.8%) | good (72.3%, 45.4%) |
| stability | unstable | stable | stable |

**Table 1.** Comparison of three hierarchical clustering algorithms. The two timings (percentages) in the speed (quality) entry are for the clustering time in second on an AMD Athlon dual-core 1.05 GHz laptop CPU (samples with silhouette value larger than 0.25) with 2000 and 500 samples, respectively.

## References

1. Sauber, N., Theisel, H., Seidel, H.P.: Multifield-graphs: An approach to visualizing correlations in multifield scalar data. IEEE Transactions on Visualization and Computer Graphics **12** (2006) 917–924
2. Qu, H., Chan, W.Y., Xu, A., Chung, K.L., Lau, K.H., Guo, P.: Visual analysis of the air pollution problem in Hong Kong. IEEE Transactions on Visualization and Computer Graphics **13** (2007) 1408–1415
3. Glatter, M., Huang, J., Ahern, S., Daniel, J., Lu, A.: Visualizing temporal patterns in large multivariate data using textual pattern matching. IEEE Transactions on Visualization and Computer Graphics **14** (2008) 1467–1474
4. Sukharev, J., Wang, C., Ma, K.L., Wittenberg, A.T.: Correlation study of time-varying multivariate climate data sets. In: Proceedings of IEEE VGTC Pacific Visualization Symposium. (2009) 161–168
5. Ankerst, M., Berchtold, S., Keim, D.A.: Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In: Proceedings of IEEE Symposium on Information Visualization. (1998) 52–60
6. Yang, J., Peng, W., Ward, M.O., Rundensteiner, E.A.: Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In: Proceedings of IEEE Symposium on Information Visualization. (2003) 105–112
7. Izenman, A.J.: Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning. 1st edn. Springer Texts in Statistics. Springer (2008)
8. Zimek, A.: Correlation Clustering. PhD thesis, Ludwig-Maximilians-Universität München (2008)
9. Pons, P., Latapy, M.: Computing communities in large networks using random walks. Journal of Graph Algorithms and Applications **10** (2006) 191–218
10. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics **20** (1987) 53–65