Correlation Study of Time-Varying Multivariate Climate Data Sets

Jeffrey Sukharev* Chaoli Wang† Kwan-Liu Ma‡ Andrew T. Wittenberg §
University of California, Davis National Oceanic and Atmospheric Administration

ABSTRACT

We present a correlation study of time-varying multivariate volumetric data sets. In most scientific disciplines, to test hypotheses and discover insights, scientists are interested in looking for connections among different variables, or among different spatial locations within a data field. In response, we propose a suite of techniques to analyze the correlations in time-varying multivariate data. Various temporal curves are utilized to organize the data and capture the temporal behaviors. To reveal patterns and find connections, we perform data clustering and segmentation using the k-means clustering and graph partitioning algorithms. We study the correlation structure of a single or a pair of variables using point-wise correlation coefficients and canonical correlation analysis. We demonstrate our approach using results on time-varying multivariate climate data sets.

Index Terms: G.3 [Probability and Statistics]: Multivariate Statistics; G.3 [Probability and Statistics]: Time Series Statistics; J.2 [Physical Sciences and Engineering]: Earth and Atmospheric Sciences

1 INTRODUCTION

Finding connections or correlations among data is one of the central themes of many scientific studies. A good example is climate research and forecasting, which has far-ranging applications to agriculture, fisheries, ecosystems, water resources, energy infrastructure, and disaster planning. Given the many complex interactions among the Earth's oceans, atmosphere, land, ice and biogeochemistry, and the sheer size of observational and climate model data sets, it is often difficult to identify which processes matter most for a particular climate phenomenon. A useful approach has been to examine correlations among different variables to identify relationships. But until recently, it has been difficult for climate scientists to perform such investigations rapidly and interactively with their large multidimensional data sets.

This paper focuses on the correlation study of time-varying multivariate data sets. Time-varying data are considered as a set of samples (e.g., voxels or blocks), each associated with a vector of representative or collective values (e.g., data value or derived quantity) over time. We refer to such a vector as a *temporal curve*. Correlation analysis thus operates on temporal curves of data samples. A temporal curve can be treated as a two-dimensional function where the two dimensions are time and data value (or derived quantity). It can also be treated as a point in the high-dimensional space (when the number of time steps taken in the study is large). In this case, to facilitate effective analysis, it is often necessary to transform temporal curve data from the original space to some other space of lower dimensionality. Clustering and segmentation of temporal curve data in the original or transformed space provides us a way to categorize and visualize data of different patterns, which reveals connection or correlation of data among different variables or at different spatial locations. In this paper, we study the variability and correlations of a single or a pair of variables using point-wise correlation coefficients and canonical correlation analysis.

We demonstrate the effectiveness of our approach using tropical oceanic data simulated by the National Oceanic and Atmospheric Administration (NOAA) Geophysical Fluid Dynamics Laboratory (GFDL) CM2.1 global coupled general circulation model [3, 19]. The model was run with its atmospheric composition, solar forcing, and land cover fixed at pre-industrial (1860) conditions, and spontaneously produces El Niño events and many other kinds of variability that scientists seek to characterize and understand. We point out that the solution we propose in this paper is not limited to climate research and it can benefit many other scientific fields where correlation study is needed.

2 RELATED WORK

Wong and Bergeron [20] gave an overview of the work in multidimensional multivariate visualization. One of the most popular statistics techniques to study correlation between multiple variables is the scatterplot matrix which presents multiple adjacent scatterplots. Each scatterplot in the matrix is identified by its row and column numbers. The limitation of scatterplot matrix is that it does not offer an observation of correlations in the native coordinates of the data.

Another popular visualization technique for finding relationships among multivariate data is the parallel coordinate [9]. It has proven to be very useful in revealing correlations among multiple variables or quantities through brushing and linking and has been extensively used in many areas of information and scientific visualization [11, 12] due to its simplicity and effectiveness. Qu et al. [13] introduced a S-shape axis representation to effectively encode directional information. Parallel coordinate, however, is not very natural for revealing spatial relationships, which are important to the analysis of climate data sets because proximity to the coastlines, the equator, cloudy regions, and oceanic jets can strongly influence the dynamical regime of the points being examined.

Sauber et al. [14] proposed to visualize correlations in 3D multifield scalar data using gradient similarity measure (GSIM) and local correlation coefficient (LCC). They developed a new interface, called *Multifield-Graph*, which visualizes a number of scalar fields and their correlations. Multifield-Graph allows users to gain an overview of the amount of correlations in all possible correlation fields and to focus on nodes with highly correlating fields. Gosink et al. [8] performed a localized correlation study where the correlation field is defined as the normalized dot product between two gradient fields from two variables. The derived correlation field was used to study variable interactions with a third variable in query-driven visualization. Qu et al. [13] adopted the standard correlation coefficient for calculating the correlation strengths between different data attributes in weather data analysis and visualization. They created a weighted complete graph where the node represents the data attribute and the weight of the edge between two nodes encodes the strength of correlation. The weighted complete graph is employed to reveal the overall correlation of all data attributes. Glatter et al. [7] used two-bit correlation to study temporal patterns in large multivariate data. Specifically, two-bit correlation

*e-mail: jsukharev@ucdavis.edu
†e-mail: wangcha@cs.ucdavis.edu
‡e-mail: ma@cs.ucdavis.edu
§e-mail: andrew.wittenberg@noaa.gov

IEEE Pacific Visualization Symposium 2009
April 20 - 23, Beijing, China
978-1-4244-4404-5/09/$25.00 ©2009 IEEE

shows four possible scenarios: two variables both changing positively, negatively, or in different directions.

In this paper, our goal is to analyze and visualize correlations among different fields or variables within slices or volumes to help scientists discover relationships and compare feedback loops among different models.

3 Temporal Curve

3.1 Time-Activity Curve

An appropriate form of data representation is needed to study the temporal aspect of time-varying data. One form of representation is the *time-activity curve* (TAC) presented by Fang et al. [5]. The basic idea of TACs is to treat each voxel in the volume as a temporal function $f(v) = (v_1, v_2, ..., v_t)$, and the source of this temporal behavior varies with a particular modality. In their work, Fang et al. matched all TACs of voxels in the volume based on certain similarity measures to identify and visualize regions with the corresponding temporal pattern. Woodring and Shen [21] utilized TACs to perform temporal clustering. By employing the wavelet transform along the time axis, they transformed data points into multiscale time series curve sets. Clustering the time curves groups data of similar activity at different temporal resolutions, which are displayed in a global time view spreadsheet.

3.2 Subsampling in Space and Time

The simplest definition of TAC uses the scalar data values at each time step as the function values. The concept of TAC can be extended from a voxel to a data block (i.e., spatial neighboring voxels) where the mean or variance value or some other quantity of the data block is used as a representative value. This treatment is useful when data sets get larger in the spatial dimensions and the users demand a quick overview. When the users examine local regions of interest, however, spatial subsampling should not be adopted. Otherwise, what are often the most interesting features (which tend to occur in thin strips near coasts and along the equator of the climate data set) would be washed out.

If the data consists of a large number of time steps, another way to get a faster response is to randomly subsample the time steps. The user can specify the subsampling factor for correlation study. For example, she could choose a large time interval for rapid interactive exploration of the overall structure, and then refine the interval when potentially interesting features are identified. This treatment also enables the users to assess the robustness of the correlation statistics, i.e., to determine whether correlations are revealing real physical relationships, or are just appearing spuriously due to coincidental covariability of fields in a short time series, by repeatedly estimating the correlations using different random subsamples of the full time series. The less the display changes between subsamples, the more robust the result.

3.3 Importance Curve

A further extension of TAC is the *importance curve* (IC) proposed by Wang et al. [18]. The IC considers the local statistics (i.e., the multidimensional histogram in the feature space) of a data block and evaluates the relative amount of information (i.e., conditional entropy) of the data block with respect to other blocks in the time sequence. In this case, the temporal function f returns the importance value of the given block, characterizing its local temporal behavior. One attractive aspect of the IC is that it can take multiple variables in the feature representation for importance value calculation, which makes it also amenable for multivariate data analysis. In this paper, we utilize both TAC and IC in correlation study, which we describe next.

4 Correlation via Clustering or Segmentation

In this paper, we present two different ways to cluster or segment data organized in the forms of temporal curves. One way is to group temporal curves of data into clusters of similar trends using the k-means clustering algorithm (Sections 4.1). This method operates directly on the data in the original high-dimensional space. On the contrary, another way to perform data clustering or segmentation is based on the data transformed into the principal component space (Sections 4.2). We utilize the normalized cut algorithm (Sections 4.3) to segment the transformed data.

4.1 K-Means Clustering

The k-means clustering is an algorithm to cluster n objects based on attributes into k partitions, $k < n$. Intuitively speaking, it attempts to find the centers of natural clusters in the data. The algorithm assumes that the object attributes form a vector space. The objective is to minimize total intra-cluster variance. The most common form of the popular k-means algorithm uses an iterative refinement heuristic called Lloyd's algorithm. Although it converges very quickly, Lloyd's algorithm could get stuck in local minima that are far from the optimal. For this reason we also consider heuristics based on local search, in which centroids are swapped in and out of an existing solution randomly (i.e., removing some centroids and replacing them with other candidates). Such a swap is accepted if it decreases the average distortion (the distortion between a centroid and a point is defined as their squared Euclidean distance); otherwise it is ignored. This hybrid k-means clustering algorithm [10] combines Lloyd's algorithm and local search by performing some number of swaps followed by some number of iterations of Lloyd's algorithm. Furthermore, an approach similar to simulated annealing is included to avoid getting trapped in local minima (refer to [10] for detail).

4.2 Principal Component Analysis

Invented by Karl Pearson in 1901, principal component analysis (PCA) is a powerful and popular tool for deriving the dominant patterns in a statistical field (e.g., a random vector $\mathbf{x}$, usually indexed by location in space). It is a vector space transform often used to reduce multidimensional data sets to lower dimensions for further analysis. Theoretically, PCA is the optimal transform for a given data in least square terms. In order to faithfully represent the data in the low-dimensional space, it is preferable that 90% of the data variances are mapped on the first two principal components. For example, our test results on the climate data set show that PCA is ideal for dimension reduction as the first principal component already describes 80-90% of data variances.

4.3 Normalized Cut

After dimension reduction using PCA, a suitable clustering method is needed so that distances between data points in the reduced dimensions can be used to generate clusters accordingly. In this paper, we utilize the normalized cut algorithm from image segmentation literature by treating data after dimension reduction (using either PCA or derived statistics) as images. We have decided to use the normalized cut due to its ability to find perceptually significant groups first before detecting smaller, less significant groups. The normalized cut takes three parameters as input: the image itself, the desired number of clusters, and the distances between image data points. We calculate these distances using two metrics: Euclidian and Manhattan.

The normalized cut, introduced by Shi and Malik [15], is a graph partitioning method that breaks a graph into segments. The algorithm represents the input image as a fully connected graph where every pixel has a link to every other pixel. It was designed to overcome outliers. Instead of looking at the value of total edge weight

connecting the two partitions A and B ($A \bigcup B = Q$), the method computes the cut cost as a fraction of the total edge connections to all nodes:

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,Q)} + \frac{cut(A,B)}{assoc(B,Q)} \quad (1)$$

where

$$assoc(A,Q) = \sum_{a,q} w(a,q), \quad assoc(B,Q) = \sum_{b,q} w(b,q) \quad (2)$$

$$a \in A, \quad b \in B, \quad q \in Q$$

Assuming the size of the input image is $n \times m$, the product of temporal curves can be represented as 1D vector Q of size $N = n \times m$. We compute the weight matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$, where $\mathbf{W}(i,j)$ represents relationship between points i and j in Q. Given the weight matrix $\mathbf{W}$ and the number of clusters c, we compute the degree matrix $\mathbf{D} = Diag(\mathbf{W}_{1N})$, where $\mathbf{W}_{1N} \in \mathbb{R}^N$ and each element is the sum of the corresponding rows in $\mathbf{W}$.

We then find the optimal eigensolution $\mathbf{Z}^*$ by solving the leading c eigenvectors using the standard eigensolver:

$$\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}}\mathbf{v} = \lambda \mathbf{v} \quad (3)$$

$$\mathbf{Z}^* = \mathbf{D}^{-\frac{1}{2}}\mathbf{V}_{[c]} \quad (4)$$

where $\mathbf{v}$ is the eigenvector and λ is the eigenvalue. The clustering results are then displayed in the principal component space, or directly on the region selected.

5 Point-Wise Correlation Analysis

5.1 Pearson Product-Moment Correlation Coefficient

The Pearson product-moment correlation coefficient is a common measure of the degree of linear correlation between two variables X and Y:

$$\rho_{XY} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{X_i - \mu_X}{\sigma_X}\right)\left(\frac{Y_i - \mu_Y}{\sigma_Y}\right) \quad (5)$$

where μ_X (μ_Y) and σ_X (σ_Y) are the mean and standard deviation of X (Y) respectively. The result obtained is equivalent to dividing the covariance between X and Y by the product of their standard deviations σ_X and σ_Y. The correlation value of ρ_{XY} ranges from -1 to 1. A correlation of 1 (-1) means that there is a perfect positive (negative) linear relationship between X and Y. A value of 0 shows that a linear model is inappropriate, i.e., there is no linear relationship between X and Y.

5.2 Self-Correlation and Cross-Correlation

In our study, the input of X and Y in Equation 5 could be two temporal curves of the same variable V at two different sampling locations. This case corresponds to the study of the *self-correlation matrix* $\mathbf{C_s}(i,j) = \rho_{i,j}$, where $1 \le i \le N$, $1 \le j \le N$, and N is the number of sampled locations considered. The self-correlation matrix tells us the relationships of the individual fields to themselves at different spatial locations. Note that $\mathbf{C_s}$ is symmetric, with all ones on the diagonal. The kth row in the self-correlation matrix gives the same correlation map as the kth column. In our implementation, we provide a user interface that allows the users to specify a reference location and observe the correlations of all other samples with respect to the reference location. This corresponds to the calculation and observation of one row or column of $\mathbf{C_s}$ at a time. In general, correlation coefficients approach 1 at locations near the reference location, since data values in the vicinity of the reference location tend to exhibit the same variations.

We could also use two temporal curves of two different variables V_a and V_b at the same or different locations as the input of X and Y in Equation 5. This case corresponds to the study of the *cross-correlation matrix* $\mathbf{C_c}(i,j) = \rho_{i,j}$, where $1 \le i \le N$, $1 \le j \le N$, and N is the number of sampled locations considered. In the cross-correlation matrix, each row is a 3D map for V_a and each column is a 3D map for V_b where the two samples are taken from different spatial locations. In contrast, the diagonal of $\mathbf{C_c}$ corresponds to the correlations where the two samples are taken from the same spatial locations. We allow the users to specify a reference location in V_a and observe the correlations of all other samples in V_b with respect to the reference location, and vice versa.

Note that the above two cases of self-correlation and cross-correlation studies are all implemented in the GPU. This allows the user to interactively change the reference location in real time and observe the correlation in the data fields accordingly.

6 Canonical Correlation Analysis

The Pearson correlation coefficient is a point-wise *local* analysis since the input only takes two local samples. In contrast, canonical correlation analysis is a *global* analysis which considers the correlation structure among all samples.

6.1 CCA and Its Relation with PCA

Canonical correlation analysis (CCA) is similar to PCA, or empirical orthogonal functions (EOFs), as it is best known in climate research community. Just as PCA is used to study the variability of a random vector $\mathbf{x}$ (i.e., what patterns account for most of the variance of the field), CCA is used to study the correlation structure of a pair of random vectors $\mathbf{x}$ and $\mathbf{y}$ (i.e., what patterns of the two fields have expansion time series that are most highly correlated).

CCA was first described by Harold Hotelling in 1936. CCA defines coordinate systems that optimally describe the cross correlation between two different data sets. Just like PCA, the optimization can be expressed as an eigenvalue problem. In this case, the eigenstructure is obtained from the product of the cross-correlation matrix between two data sets and its transpose. The strengths of CCA are its ability to operate on the full fields of information and to objectively define the most highly related patterns of both data sets [1].

CCA and PCA share similar objectives and mathematics [17]. One interpretation of the first principal component $\mathbf{v}_1$ of $\mathbf{x}$ is that $\mathbf{x}^T\mathbf{v}_1$ is the linear combination of elements of $\mathbf{x}$ with the greatest variance. The second principal component $\mathbf{v}_2$ is spatially orthogonal to $\mathbf{v}_1$, and provides the linear combination $\mathbf{x}^T\mathbf{v}_2$ with greatest variance that is temporally uncorrelated with $\mathbf{x}^T\mathbf{v}_1$, and so on. The objective of CCA is to find a pair of patterns $\mathbf{w}_{x1}$ and $\mathbf{w}_{y1}$ so that the correlation between linear combinations $\mathbf{x}^T\mathbf{w}_{x1}$ and $\mathbf{y}^T\mathbf{w}_{y1}$ is maximized. A second pair of patterns $\mathbf{w}_{x2}$ and $\mathbf{w}_{y2}$ is found so that $\mathbf{x}^T\mathbf{w}_{x2}$ and $\mathbf{y}^T\mathbf{w}_{y2}$ are the most strongly correlated linear combinations of $\mathbf{x}$ and $\mathbf{y}$ that are not correlated with $\mathbf{x}^T\mathbf{w}_{x1}$ and $\mathbf{y}^T\mathbf{w}_{y1}$, and so on.

In practice, we can think of CCA as a static way to explore the correlations between two different data fields. Instead of solving the eigenvalue problem of the covariance matrix as in PCA, CCA solves the cross-correlation matrix. The resulting CCA patterns maximize the correlation between the domain vectors. Whether or not CCA or PCA patterns should be used as the preferred presentation depends on the data and question posed [4].

6.2 Calculation of CCA Patterns

In the following, we explain in detail how to calculate CCA patterns. First of all, two space-time data fields are represented as two 2D matrices. Each matrix has n rows representing time steps and p columns representing spatial locations (i.e., grid points or voxels). For 3D volume data, it is often the case that the number of spatial locations p is much larger than the number of time steps n, which causes problems in CCA calculation. To get around this issue, we

reduce the number of spatial locations to a more manageable size. This is achieved with the use of PCA [17], which reduces the number of spatial locations from p to r, where r is the desired principal components.

In PCA interpretation, the columns of the input matrix $\mathbf{X}$ represent different variables or spatial grid points, and rows represent observations. PCA is computed by the following function from the `MatLab` Toolbox for Dimensionality Reduction [16]:

$$[\mathbf{C}_{p\times r}, \mathbf{S}_{n\times r}, \mathbf{L}_{r\times 1}] = \text{pca}\,(\mathbf{X}_{n\times p}, r) \tag{6}$$

Here we use the subscripts of input and output matrices to denote their respective dimensions. The PCA function takes one of the two data fields as input. The output of the function are: the coefficient matrix $\mathbf{C}_{p\times r}$ where each column represents a p-element principal component pattern or spatial map (note that only the leading r of them are requested); the score matrix $\mathbf{S}_{n\times r}$ which contains the n-element projection of time series associated with the r principal component patterns; and the 1D latent matrix $\mathbf{L}_{r\times 1}$ which records the variances associated with each of the r projected time series.

CCA is calculated by the following function from the `MatLab` Statistical Toolbox:

$$[\mathbf{A}1_{r\times d}, \mathbf{A}2_{r\times d}] = \text{canoncorr}\,(\mathbf{S}1_{n\times r}, \mathbf{S}2_{n\times r}, d) \tag{7}$$

where the input matrices $\mathbf{S}1_{n\times r}$ and $\mathbf{S}2_{n\times r}$ come from the output of the PCA function for each of two data fields. d is the number of desired CCA patterns. The output of the CCA function are matrices $\mathbf{A}1_{r\times d}$ and $\mathbf{A}2_{r\times d}$, which represent canonical patterns for the PCA-transformed input data fields $\mathbf{S}1_{n\times r}$ and $\mathbf{S}2_{n\times r}$, respectively.

Finally, CCA patterns are transformed back to the original data space and displayed. The transformation is accomplished using the following equation:

$$\mathbf{P}1_{p\times d} = \mathbf{C}1_{p\times r} * \mathbf{A}1_{r\times d} \tag{8}$$

$$\mathbf{P}2_{p\times d} = \mathbf{C}2_{p\times r} * \mathbf{A}2_{r\times d} \tag{9}$$

where each row in $\mathbf{P}1_{p\times d}$ and $\mathbf{P}2_{p\times d}$ represents a spatial map of the CCA patterns, which is displayed in the original data space.

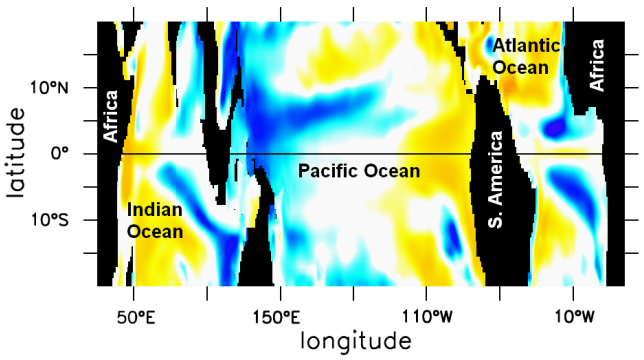


Figure 1: A slice of the temperature self-correlation field normal to the z axis. The reference location is at (140°W, 0°, 0m). Blue and yellow are for negative and positive linear correlations, respectively (white for no linear correlation). Continents, oceans, longitudes, and latitudes are labeled.

7 Results

We performed the correlation study on the climate data set provided by the NOAA scientists. The equatorial upper-ocean climate data set covers 20°S to 20°N over a period of 100 years, which is sufficient for us to prototype a correlation browser. The data are sampled at one time step per month and there are 1200 time steps in total. The spatial dimension of the data is $360 \times 66 \times 27$, with the x axis for longitude, the y axis for latitude, and the z axis for depth. In Figure 1, we show a labeled slice of the temperature self-correlation field normal to the z axis. We studied the temperature and salinity variables in our experiments. The timing performance on a 2.33GHz Intel Xeon processor with 4GB main memory is listed in Table 1.

TAC clustering (23760 blocks)	1200 time steps	191s
IC clustering (3960 blocks)	1200 time steps	20s
normalized cut	105 time steps	37.56s
CCA	310 time steps	2.84s

Table 1: The timing performance of the correlation analysis. Note that the clustering was performed on all data blocks of the entire volume while normalized cut and CCA were performed on a selected 2D region or a slice of the data due to the memory constraint.

Throughout this section, we give examples on how to detect ENSO (El Niño-Southern Oscillation), a global coupled ocean-atmosphere phenomenon including El Niño and La Niña using the correlation study. Note that the climate scientists helped us interpret the results reported in this section. Due to the complexity of the data in the spatial, temporal, and variable domain, the scientists are much more interested in visualization tools that allow them to explore the correlation structure. Our goal is to enable scientists to investigate the multifaceted nature of their data.

7.1 Clustering and Segmentation Result

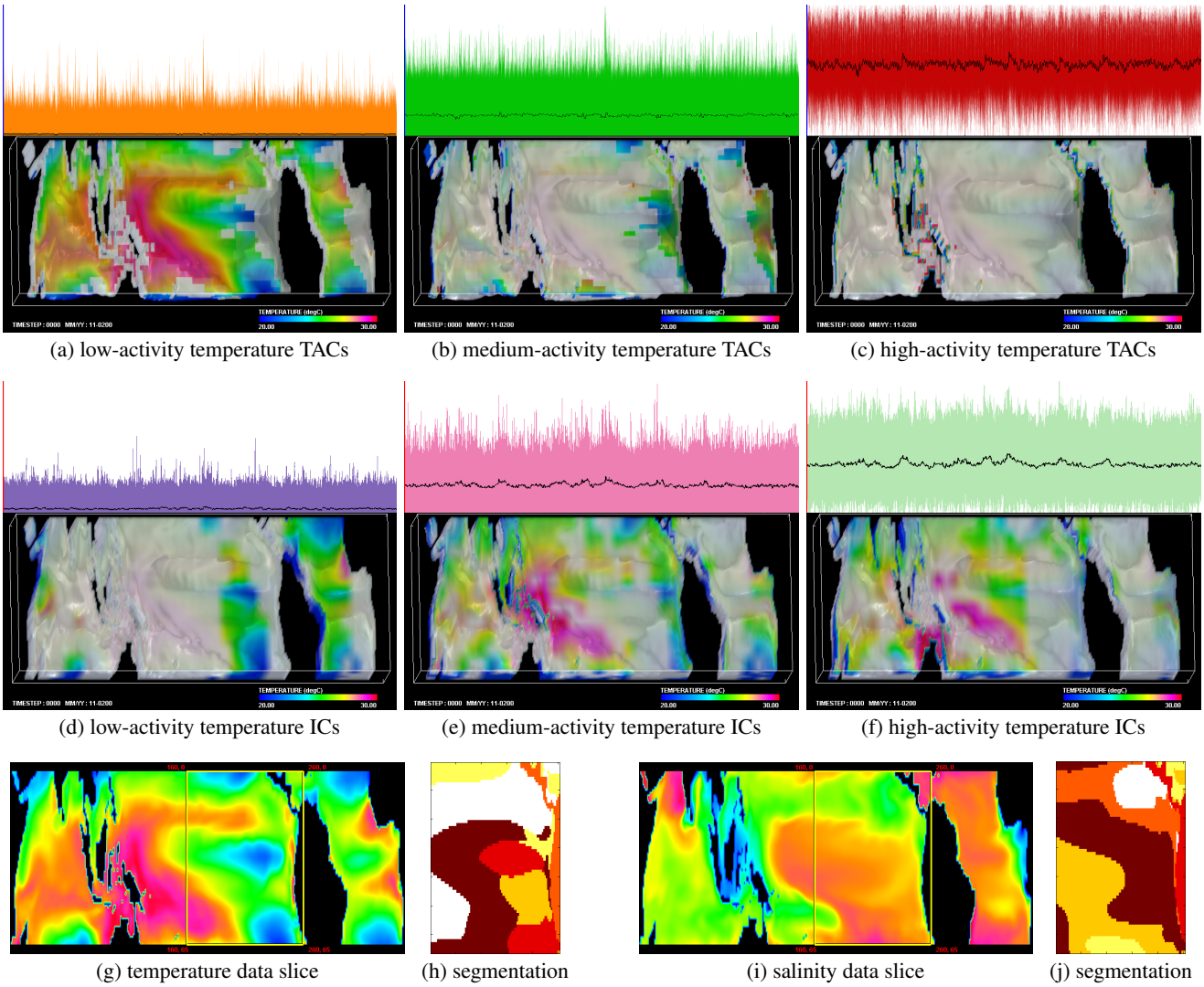
Figure 2 (a)-(f) show the clustering results with the TACs and ICs. The data were partitioned into $3 \times 3 \times 3$ for TAC clustering and $9 \times 6 \times 3$ for IC clustering. For TAC clustering, the variances of the data blocks were used. For IC clustering, the feature vector included both temperature and salinity values in the multidimensional histogram calculation. Instead of clustering all time steps together, we clustered time segments (each with 12 time steps) and then matched clusters by sorting the end points of the centroids in neighboring time segments and making correspondence according to their orders. Such a treatment generally reduces the computation time and improves the average distortion.

Figure 2 (a)-(c) show the clustering results of the TACs. We note that voxels corresponding to the continents were not included in the variance calculation of data blocks. It is interesting to observe that the cluster with the highest variance in temperature is mostly distributed along the coastlines (Figure 2 (c)). Figure 2 (d)-(e) show the clustering results of the ICs. The data are categorized according to the degree of temporal changes in both temperature and salinity values. Essentially, the ICs measure the "unusualness" of the data blocks with respect to other neighboring blocks in the time series. From Figure 2 (a), (b), (d), and (e), we can infer that most of the regions in the Atlantic Ocean do not exhibit dramatic temporal changes. Figure 2 (g)-(j) show the segmentation results of the 2D region on the sea surface data slice over 105 time steps. The normalized cut algorithm was applied to the selected data transformed into the principal component space.

7.2 Point-Wise Correlation Result

The point-wise correlation computation was implemented in the GPU, which allows the user to explore the correlation interactively by dragging the reference location around at runtime. Time steps are loaded into textures by stacking them along the z axis. The fragment program retrieves values of sample locations over time through texture lookups. The calculation of the Pearson correlation coefficient follows after the values are filled for each fragment.

Figure 3 shows the self-correlation of the temperature and salinity variables respectively. The reference location is at (140°W, 0°,



(a) low-activity temperature TACs (b) medium-activity temperature TACs (c) high-activity temperature TACs

(d) low-activity temperature ICs (e) medium-activity temperature ICs (f) high-activity temperature ICs

(g) temperature data slice (h) segmentation (i) salinity data slice (j) segmentation

Figure 2: First and second rows: k-means clustering the TACs (a)-(c) and the ICs (d)-(f) of the 1200 time steps climate temperature data set. Left to right: three clusters of temporal curves with increasing temporal activity are shown. The corresponding rendering at the first time step (indicated by the time lines in their temporal curves) are also displayed. Each cluster is highlighted with high saturated colors while the rest of data are rendered with low saturated colors for the context. Centroids of clusters are displayed in black on top of the temporal curves. Third row: the segmentation result corresponding to the 2D region selected over 105 time steps. The first time steps of temperature and salinity variables are shown in (g) and (i), respectively. Six clusters denoted by different colors are shown in (h) and (j), respectively.

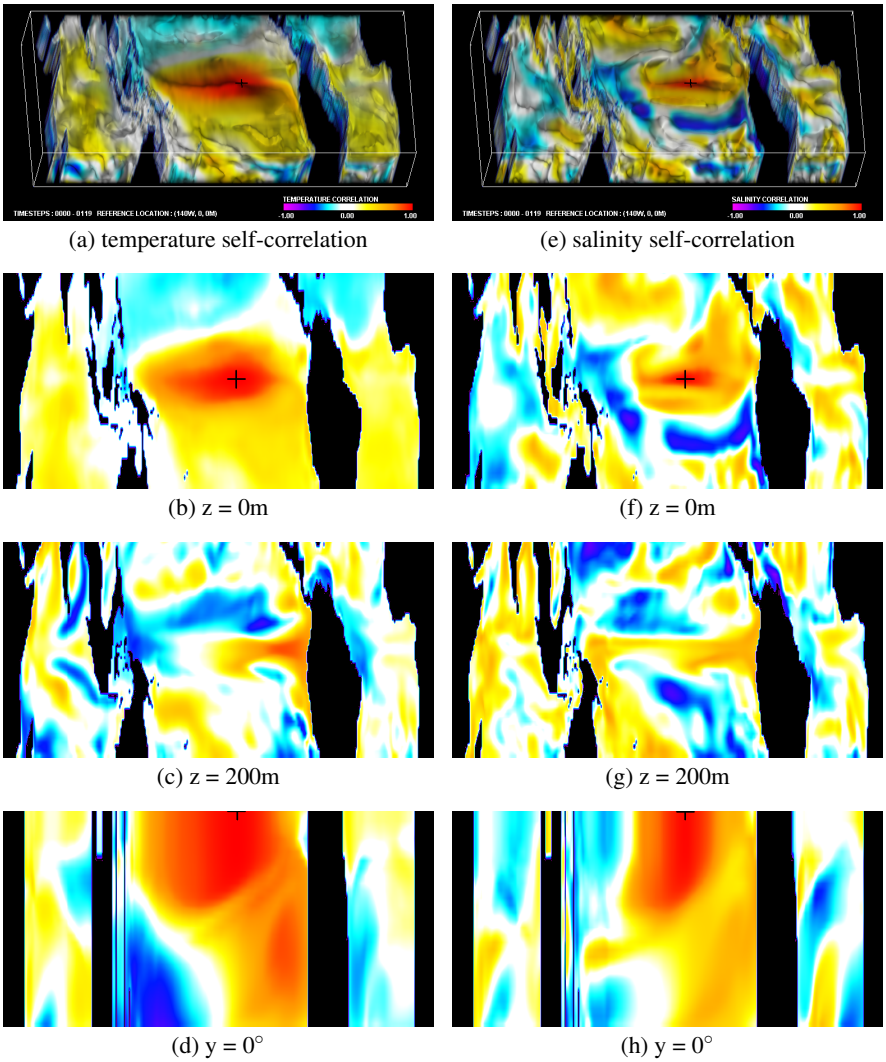
0m), which corresponds to the surface in the eastern equatorial Pacific (marked with a cross sign). Three slices with z = 0m, z = 200m, and y = 0° are also shown. Colors vary from purple to blue to cyan for negative correlations, white for zero correlations, and yellow to orange to red for positive correlations. Using the same color map, Figure 4 shows the cross-correlation of the temperature and salinity variables.

The reference location (140°W, 0°, 0m) is a great indicator of El Niño and La Niña. As we can see during El Niño, we get a warmer ocean surface especially over the equatorial Pacific, and the surface waters get fresher in the Indian Ocean and equatorial West Pacific (Figure 3 (b) and Figure 4 (b)); at 200m the equatorial temperatures and salinities change in concert (Figure 3 (c) and Figure 4 (c)); and the temperatures and salinities have a secondary extrema at depth (Figure 3 (d) and Figure 4 (d)). All of these results are scientifically interesting.

Finally, the right column of Figure 4 show the cross-correlation of temperature and salinity values at the same locations. This is potentially useful for determining contributions to seawater density, which along with sea level and winds helps to control the ocean currents. Positive correlations indicate regions where the density impact of salinity variations (saltier = denser) counteracts that of temperature variations (warmer = lighter). The negative correlations in the western equatorial Pacific may result from El Niño, which shifts thunderstorms eastward from Indonesia and pours fresh rainwater on the surface as the equatorial ocean warms.

7.3 CCA Result

Figure 5 shows the CCA result of using the sea surface slice of the temperature and salinity data over 310 time steps. In this experiment, we used the anomaly values (i.e., data values depart from the normal seasonal cycle) rather than data values so that the impact of the seasonal cycle is eliminated. To obtain robust CCA patterns, we increased the number of principal components r in Equation 6 until the CCA patterns become insensitive to further increase of r. In the figure, the value of r we used is 15. The first two leading CCA



(a) temperature self-correlation (e) salinity self-correlation

(b) z = 0m (f) z = 0m

(c) z = 200m (g) z = 200m

(d) y = 0° (h) y = 0°

Figure 3: Volume and slice rendering of the self-correlation map. The first 120 time steps of the climate data set are used. Left column: temperature variable. Right column: salinity variable. The reference location (140°W, 0°, 0m) is indicated with a cross sign.

patterns are shown where the correlation value p is scaled using the following equation:

$$p_{scaled} = \frac{p}{\max(|p_{\min}|, |p_{\max}|)} \tag{10}$$

where $p_{\min}$ and $p_{\max}$ are the minimum and maximum correlation values.

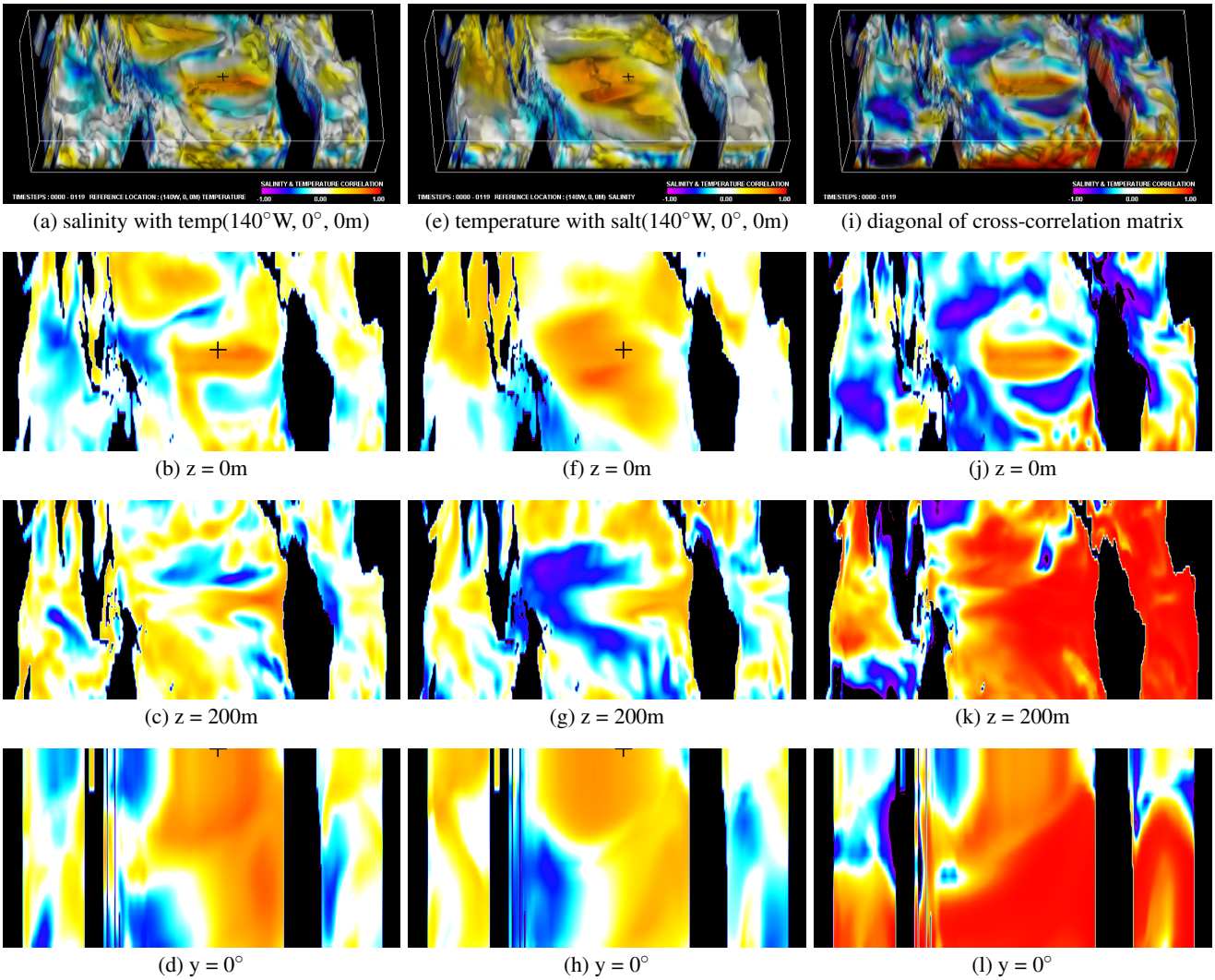
Given that ENSO is the main source of large-scale correlations in this climate data, the CCA patterns indicate what happens during an El Niño event, when the equatorial Pacific is warm. When such an event happens, the atmospheric convection (thunderstorms) follow the warm water eastward, moving from Indonesia into the western Pacific, giving more fresh rainwater input (and a fresher surface ocean) in the west Pacific. The Intertropical Convergence Zone (ITCZ) in the central Pacific along 10-15°N also shifts equatorward, giving more rain (and a fresher surface) near the equator, and less rain (saltier surface) farther north.

Note that since CCA is a linear technique, we can arbitrarily flip the sign of the resulting patterns (as long as we flip the sign for both variables at once). In this case, we will have the correlation that corresponds to a La Niña event, with the cold temperature (blue) in the central Pacific and the salty water (red) in the west Pacific.

8 Conclusion and Future Work

In this paper, we presented correlation analysis and visualization of the climate data. Our approach provides a general and powerful way to explore correlations in time-varying multivariate data. Immediate extensions of this work which we would like to explore include adding covariances, regressions, and partial regressions to our point-to-point browser. We would also consider replacing probing point with probing cube, synchronizing multiple views, and studying lag correlations, cross-model correlations, as well as observation-forecast correlations. Besides CCA, we could look at maximal covariance analysis (which asks what patterns of one field has the largest covariance with another field) and redundancy analysis (which asks what patterns of one field predict the maximum variance of another field). With these additions, all kinds of applications and multivariate extensions can be studied: we could rapidly investigate how atmospheric winds are related to ocean temperatures; how atmospheric moisture links to subsurface soil water; and how biological productivity links to ocean currents etc.

In the future, we would explore using perceptual color mapping schemes [2] for a more optimal color scale instead of the rainbow color map. Our current implementation of the normalized cut algorithm and the calculation of CCA rely on `MatLab`, which has its



(a) salinity with temp(140°W, 0°, 0m) (e) temperature with salt(140°W, 0°, 0m) (i) diagonal of cross-correlation matrix

(b) z = 0m (f) z = 0m (j) z = 0m

(c) z = 200m (g) z = 200m (k) z = 200m

(d) y = 0° (h) y = 0° (l) y = 0°

Figure 4: Volume and slice rendering of the cross-correlation map with temperature and salinity variables. The first 120 time steps of the climate data set are used. Left column: cross-correlation of salinity values with temperature value at (140°W, 0°, 0m). Middle column: cross-correlation of temperature values with salinity value at (140°W, 0°, 0m). Right column: cross-correlation of temperature and salinity values at the same locations, which correspond to the diagonal of the cross-correlation matrix.

strict memory limitation. We would like to reimplement it in C++ so that larger data input can be handled. For the normalized cut algorithm, we also plan to incorporate the Nyström method [6] so that we can extrapolate the complete solution clusters using only a small fraction of samples instead of every single data item. This will significantly reduce the algorithm's memory requirements. Performance speed up can also be sought using the GPU implementation. Integrating all these techniques with a user interface, we aim to deliver a highly interactive correlation analysis and visualization tool to the climate scientists.

Acknowledgements

This research was supported in part by the U.S. National Science Foundation through grants CNS- 0716691, OCI-0325934, OCI-0749217, CNS-0551727, CCF-0811422, CCF-0808896, OCI-0749227 and OCI-049321, and the U.S. Department of Energy through the SciDAC program with Agreement No. DE-FC02-06ER25777 and DE-FG02-08ER54956. The `MatLab` toolbox for dimensionality reduction is courtesy of L. J. P. van der Maaten at Universiteit Maastricht in the Netherlands. We thank the reviewers for their helpful comments.

References

[1] T. P. Barnett and R. Preisendorfer. Origins and levels of monthly and seasonal forecast skill for United States surface air temperatures determined by canonical correlation analysis. *Monthly Weather Review*, 115(9):1825–1850, 1987.

[2] D. Borland and R. M. Taylor II. Rainbow color map (still) considered harmful. *IEEE Computer Graphics and Applications*, 27(2):14–17, 2007.

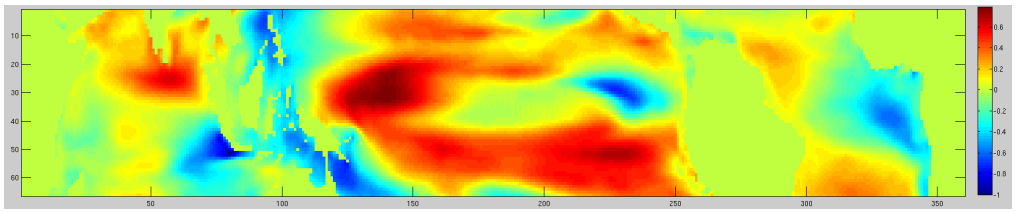
[3] T. L. Delworth, et al. GFDL's CM2 global coupled climate models, Part I: Formulation and simulation characteristics. *Journal of Climate*, 19(5):643–674, 2006.

[4] D. Dommenget. *An Introduction to Statistical Analysis in Climate Research*. Leibniz-Institut für Meereswissenschaften, July 2008.

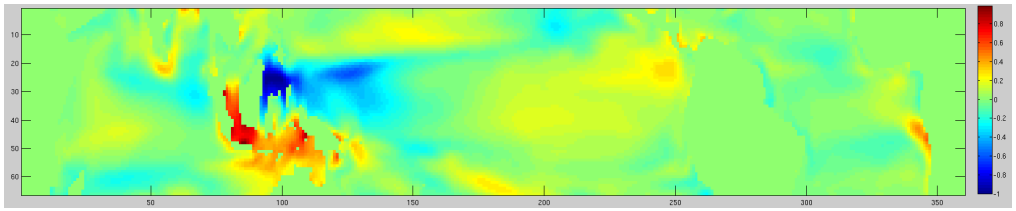
[5] Z. Fang, T. Möller, G. Hamarneh, and A. Celler. Visualization and exploration of time-varying medical image data sets. In *Proceedings of Graphics Interface 2007*, pages 281–288, 2007.

[6] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.

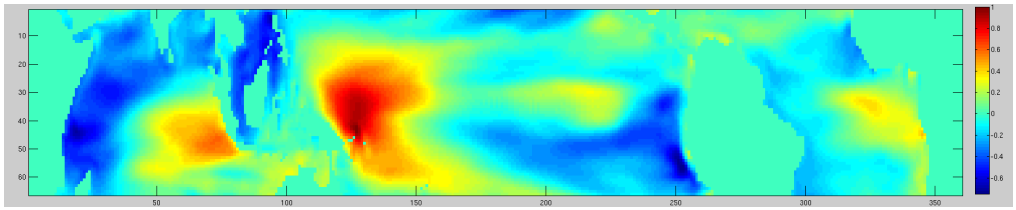
[7] M. Glatter, J. Huang, S. Ahern, J. Daniel, and A. Lu. Visualizing temporal patterns in large multivariate data using textual pattern match-



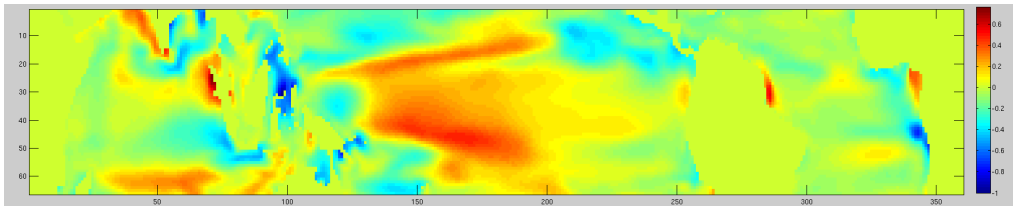
(a) the first leading CCA pattern of temperature



(b) the first leading CCA pattern of salinity



(c) the second leading CCA pattern of temperature



(d) the second leading CCA pattern of salinity

Figure 5: The CCA result with the sea surface slice of the temperature and salinity data over 310 time steps. The first two leading CCA patterns (i.e., the first and second columns of matrices $\mathbf{A}1_{r\times d}$ and $\mathbf{A}2_{r\times d}$) in Equation 7 are displayed. The displayed CCA patterns help scientists detect a La Niña event.

ing. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1467–1474.

[8] L. Gosink, J. C. Anderson, E. W. Bethel, and K. I. Joy. Variable interactions in query driven visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1000–1007, 2007.

[9] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.

[10] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. A local search approximation algorithm for k-means clustering. In *Proceedings of ACM Symposium on Computational Geometry*, pages 10–18, 2002.

[11] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[12] E. B. Lum and K.-L. Ma. Lighting transfer functions using gradient aligned sampling. In *IEEE Visualization Conference 2004*, pages 289–296, 2004.

[13] H. Qu, W.-Y. Chan, A. Xu, K.-L. Chung, K.-H. Lau, and P. Guo. Visual analysis of the air pollution problem in Hong Kong. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1408–1415, 2007.

[14] N. Sauber, H. Theisel, and H.-P. Seidel. Multifield-graphs: An approach to visualizing correlations in multifield scalar data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):917–924, 2006.

[15] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[16] L. J. P. van der Maaten. An introduction to dimensionality reduction using Matlab. Technical report, MICC 07-07, Universiteit Maastricht, July 2007.

[17] H. von Storch and F. W. Zwiers. *Statistical Analysis in Climate Research*. Cambridge University Press, 2002.

[18] C. Wang, H. Yu, and K.-L. Ma. Importance-driven time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1547–1554, 2008.

[19] A. T. Wittenberg, A. Rosati, N.-C. Lau, and J. J. Ploshay. GFDL's CM2 global coupled climate models. Part III: Tropical Pacific climate and ENSO. *Journal of Climate*, 104(5):698–722, 2006.

[20] P. C. Wong and R. D. Bergeron. 30 years of multidimensional multivariate visualization. In *Scientific Visualization, Overviews, Methodologies, and Techniques*, pages 3–33, 1997.

[21] J. Woodring and H.-W. Shen. Multiscale time activity data exploration via temporal clustering visualization spreadsheet. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):123–137, 2009.