Vision, Modeling, and Visualization (2012), pp. 1–8
Thorsten Grosch, Bernhard Preim, Holger Theisel, Klaus-Dietz Tönnies (Eds.)

# Markov Chain Driven Multi-Dimensional Visual Pattern Analysis with Parallel Coordinates

Zhao Geng, James Walker and Robert S.Laramee

Visual Computing Group, Computer Science Department, Swansea University, cszg, csjames, r.s.laramee@swansea.ac.uk

**Abstract**

*Parallel coordinates is a widely used visualization technique for presenting, analyzing and exploring multi-dimensional data. However, like many other visualizations, it can suffer from an overplotting problem when rendering large data sets. Until now, quite a few methods are proposed to discover and illustrate the major data trends in cluttered parallel coordinates. Among them, frequency-based approaches using binning and histograms are widely adopted. The traditional binning method, which records line-segment frequency, only considers data in a two-dimensional subspace, as a result, the multi-dimensional features are not taken into account for trend and outlier analysis. Obtaining a coherent binned representation in higher dimensions is challenging because multidimensional binning can suffer from the curse of dimensionality. In this paper, we utilize the Markov Chain model to compute an n-dimensional joint probability for each data tuple based on a two-dimensional binning method. This probability value can be utilized to guide the user for selection and brushing. We provide various interaction techniques for the user to control the parameters during the brushing process. Filtered data with a high probability measure often explicitly illustrates major data trends. In order to scale to large data sets, we also propose a more precise angular representation for angular histograms to depict the density of the brushed data trends. We demonstrate our methods and evaluate the results on a wide variety of data sets, including real-world, high-dimensional biological data.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: —

## 1. Introduction

Parallel coordinates, introduced by Inselberg and Dimsdale [ID90], is a widely used visualization technique for exploring large, multi-dimensional data sets. It is powerful in revealing a wide range of data characteristics such as different data distributions and functional dependencies. However, one of the limitations of parallel coordinates is visual clutter caused by rendering more polylines than available pixels. The overlapped lines often obscure the underlying patterns of the data, especially in areas with high data density.

One of the promising algorithms for discovering principal data trends for large data sets in parallel coordinates is based on data frequency [AdOL04, BBP08, NH06]. With these approaches, data is sometimes aggregated and filtered by means of binning. Due to the curse of dimensionality, most frequency-based approaches adopt a two-dimensional bin map which stores frequency of line segments between adjacent axes. A joint histogram is then rendered based on this bin map. A clustering or outlier detection method is limited to a two-dimensional subspace and the multi-dimensional features are not considered. Due to the dependencies inherent within multidimensional parallel coordinates, we need to discover and summarize the patterns which can propagate through n-dimensional space, as opposed to being limited to a two-dimensional subspace.

In this paper, we develop a probability model to guide the user to brush a subset of the data items which can represent major and minor trends in a data set. To achieve this, a weight value, which is determined by the joint probability of the n-dimensional data features, is assigned to each polyline. A polyline with a higher probability value implies that a given data tuple is part of the principal trend, whereas a lower probability value implies there are few similar patterns present. In order to compute a joint probability value,

we introduce the Markov Chain model [Rab89], which can be constructed and implemented using the binning method.

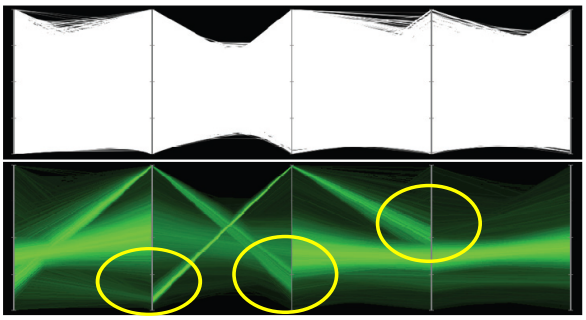

**Figure 1:** *This figure shows the line-based histogram for the remote sensor data from the paper [NH06]. This method is able to capture the local data trend in two-dimensional subspace, rather than global trend in higher dimensions. Highlighted in yellow are the discontinuous patterns due to the two-dimensional joint histogram representation.*

We demonstrate our techniques on real world n-dimensional marine biology data in addition to several well-known data sets. The rest of this paper is organized as follows: Section 2 discusses the previous work related to our approach. Section 3 demonstrates the key ideas to compute a joint probability value for a data sample using Markov chain. Once the probability values are obtained, they can be presented by a histogram or scatterplot, as discussed in Section 4. We also provide several interaction techniques for the user, such as a brushing method based on a probability value which can decompose the original data set into principal major and minor trends. Section 5 evaluates our proposed method based on a comparison with other well-known visualization techniques for large data sets. Section 7 describes a case study with respect to a real world marine biology data set. Section 8 wraps up with the conclusion.

## 2. Related Work

One of the ways to represent a major data trend in parallel coordinates is based on data frequency. With this approach, the data is often aggregated and filtered by binning. In general, binning is the process of computing the number of values falling in a given interval and storing them in a bin map. Data frequency can then be visually represented by a histogram.

In parallel coordinates, bin maps can either be line-segment based which store the frequency of the line segments connecting adjacent axes, or point based which store the frequency of data points along each axis [AdOL04]. Novotny and Hauser develop a focus+context visualization using binned parallel coordinates [NH06]. Binned parallel coordinates are used for context views, while the traditional polyline-based parallel coordinates are used for focus views. However, for the binned parallel coordinates, uniform, equal-sized histogram bins may not allow for finer-resolution views of the data. Ruebel et al. [RK08] extend Novotny and Hauser's work, and propose adaptive histogram bins which use a higher resolution in areas with high data density. Their adaptive binning is able to represent general data trends more accurately. Blaas et al. [BBP08] optimize the data preprocessing for the binning method with respect to the data storage, histogram equalization and quantization. This facilitates fast exploration for large data sets.

Because the bin map used in previous approaches is based on neighboring dimensions, it inevitably introduces discontinuous patterns across multiple dimensions. What we obtain from these binning methods are the clusters and outliers in a one or two-dimensional subspace. We are unable to visualize n-dimensional data as a coherent feature, as shown in Figure 1. In Feng [FKLT10]'s work, this discontinuity between every two dimensions is regarded as uncertainty and Kernel Density Estimation (KDE) is used to enhance such information. However, the uncertainty they consider is also based on a two-dimensional subspace. A possible solution to address this problem is to build a truly n-dimensional bin map. However, as the number of dimensions increases, the total number of required bins grows exponentially, which is $k^n$ for a data set with $n$ dimensions and $k$ bins. This can cause enormous memory demands even for a small number of intervals. In this paper, we propose a novel approach to aggregate n-dimensional data tuples using a probability model based on a line-based binning method. A two-dimensional bin map between the neighboring axes is constructed to compute a transition probability in our Markov Chain Model [Rab89]. The main contribution of our work is that we consider the multidimensional features of each polyline, not only in two dimensional subspace. In addition, based on the probability model, we can present the principal trends at various levels of detail by selection and filtering. The user is able to fine-tune different parameters of the Markov model to obtain both major and minor data trends.

**Clustering:** The ultimate goal of our work is to improve the previous line-based histogram for multidimensional pattern discovery, it focuses on data extraction and filtering for visual data pattern discovery, rather than data clustering and classification in the data space. Therefore, our technique is not the same as the traditional clustering methods, such as hierarchical clustering [FWR99] or K-Means clustering. In our approach a data trend or cluster is implicitly revealed in the visual space, but not computed in the data space.

**Outlier Detection:** Our method is not explicitly designed for outlier detection. However it can be used to detect and visualize outliers in higher dimensions using our joint probability distribution. The traditional density based outlier detection method, such as n-dimensional Kernel Density Estimation (KDE) which is based on a continuous density function, is computationally expensive especially when the size

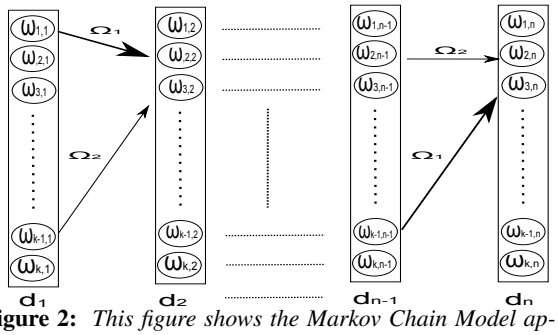


**Figure 2:** *This figure shows the Markov Chain Model applied in parallel coordinates. Each vertical axis is treated as one time step and is divided into several bins or states. The thickness of the arrow for each transition, such as $\Omega_1$ and $\Omega_2$, depicts the joint probability value.*

and dimension of a data set is high. Whereas in comparison our method with a complexity $O(nm)$ does not suffer from this problem, where $n$ is the number of dimensions and $m$ is the number of data items. Because a probability value generated by our method for each data sample indicates the number of similar patterns to it, a very low probability potentially suggests an outlier which is numerically distant from other data items.

## 3. Fundamentals

In this section, we will explain the key concepts behind our approach. Section 3.1 demonstrates the Markov Chain model developed for parallel coordinates. Based on this model, each multidimensional data tuple can be assigned a joint probability value. In order to quickly compute such a probability value, we pre-compute a transition probability matrix and store it in an external file.

### 3.1. Markov Chain Model

In this section, we will explain the key concepts behind our approach. Given a data set $X = \left\{\mathbf{x}_i = (x_{i,1}, x_{i,2}, ..., x_{i,n})^T | 1 \leq i \leq m\right\}$ with $m$ items of $n$ dimensions, the binning method converts the original data into a frequency-based representation by dividing the data space into $k$ multidimensional intervals, namely bins. The key idea behind our approach is to compute a joint probability value for each data item $\mathbf{x}_i = (x_{i,1}, x_{i,2}, ..., x_{i,n})^T$. Then we can introduce the Markov Chain to compute the joint probability of multidimensional data. A Markov Chain [Rab89] is a stochastic process that undergoes transitions from one state to another in a chainlike manner. The first-order Markov Chain defines that the current state depends only on the previous state and not on the entire past. In order to formulate the probability model, we firstly construct one-dimensional binning for every data dimension. If the data in each attribute is divided into $k$ intervals, then in total we will need $kn$ bins for all attributes, which is denoted as $S = \left\{\Omega_i = (\omega_{1,i}, \omega_{2,i}, ..., \omega_{k,i})^T | 1 \leq i \leq n\right\}$, where $\omega_{i,j}$ represents the $i^{th}$ bin in axis $j$. For every item $x_{i,j}$ in the data space, it can be converted to the bin membership by the function $\phi(x_{i,j})$, which returns to the bin index that the data item $x_{i,j}$ belongs to.

To formulate our Markov Chain model, each bin $\omega_{i,j}$ can be interpreted as a state. A list of data dimensions, $D : d_1, d_2, ..., d_n$, can be treated as a temporal sequence. Every data tuple or polyline forms a sequence of transitions from one state to another over a series of time steps. Computing a probability of a data tuple $x_i = (x_{i,1}, x_{i,2}, ..., x_{i,n})^T$ can be transformed to computing the probability of the list of states where this data tuple flows to, namely $\Phi_i = (\phi(x_{i,1}), \phi(x_{i,2}), ..., \phi(x_{i,n}))^T$, this Markov process is shown in Figure 2. A joint probability of each bin tuple $\Phi_i$ can be defined as:

$$P(\Phi_i) = P_0(\phi(x_{i,1})) \prod_{t=2}^{n} P(S_t = \phi(x_{i,t}) | S_{t-1} = \phi(x_{i,t-1})) \tag{1}$$

where $P_0(\phi(x_{i,1}))$ is a stationary probability value, which can be represented by the data frequency within this bin.

In order to improve the numerical stability when using limited precision floating point numbers for product computation in equation (1), we take the logarithm of both sides. This can be defined as:

$$\begin{aligned} ln(P(\Phi_i)) = ln(P_0(\phi(x_{i,1}))) + \\ \sum_{t=2}^{n} ln(P(S_t = \phi(x_{i,t}) | S_{t-1} = \phi(x_{i,t-1}))) \end{aligned} \tag{2}$$

Before defining the transition probability, we construct the two-dimensional binning which stores the line segment frequency between every pair of axes. Based on this bin map, we are able to build a $kn \times kn$ stochastic transition matrix, where $k$ is the number of intervals and $n$ is the number of dimensions. Each element of this matrix can be defined as:

$$P(S_t = \omega_{i,j} | S_{t-1} = \omega_{u,v}) = \begin{cases} \frac{\|\omega_{i,j} \cap \omega_{u,v}\|}{\|\omega_{u,v}\|(n-1)} & if\ j \neq v \\ 0 & otherwise \end{cases} \tag{3}$$

where $\|\omega_{i,j} \cap \omega_{u,v}\|$ represents the number of common data items that both bins $\omega_{i,j}$ and $\omega_{u,v}$ share, i.e, the number of line segments joining the two bins between the axes $j$ and $v$. $\|\omega_{u,v}\|$ is the number of data items in the bin $\omega_{u,v}$. $n$ is the number of dimensions. If two states are from the same axis ($j = v$), then the transition probability is zero. In our model, we only consider the case ($j \neq v$) where the transition probability is the normalized conditional probability of state $\omega_{i,j}$ over state $\omega_{u,v}$.

Since a state transition probability matrix has to be a

stochastic matrix, the sum of each row of the matrix has to be one. Based on equation (3), we can prove that our transition probability matrix satisfies $\sum_{v=1}^{kn} p_{u,v} = 1$, for all $1 \leq u \leq kn$, where $p_{u,v}$ represents a matrix element. We take the first row of the matrix as an example. The transition probabilities in this row can be separated into two parts, one contains the states in the same dimension as $\omega_{1,1}$, the other contains the states in different dimensions:

$$\sum_{u=1}^{kn} p_{1,u} = \sum_{u=1}^{k} p_{1,u} + \sum_{u=k+1}^{kn} p_{1,u} \tag{4}$$

Based on equation (3), the first part of equation (4), namely $\sum_{u=1}^{k} p_{1,u}$ equals zero. The second part of equation (4) computes the sum of the transition probabilities between the state $\omega_{1,1}$ and all states in the other $(n-1)$ dimensions. The part $\sum_{u=k+1}^{kn} p_{1,u}$ equals to:

$$\sum_{e=2}^{n} \sum_{u=(e-1)k+1}^{ek} \frac{\|\omega_{u,e} \cap \omega_{1,1}\|}{\|\omega_{1,1}\|(n-1)}) =$$

$$\sum_{e=2}^{n} \frac{\sum_{u=(e-1)k+1}^{ek} \|\omega_{u,e} \cap \omega_{1,1}\|}{\|\omega_{1,1}\|(n-1)} = \sum_{e=2}^{n} \frac{\|\omega_{1,1}\|}{\|\omega_{1,1}\|(n-1)} = 1 \tag{5}$$

Then the results in equations (4) equals unity. In our paper, the probability value is mainly used for ranking. According to equation (2), we learn that the degree of probability value for each data sample indicates the number of patterns which are similar to it. If two data samples are passing through a similar set of states or bins in a Markov Chain, then we say these two data samples have a similar profile in multidimensional space.

The transition probability of the Markov Chain is based on the line-segment frequency between the bins in neighboring axes, namely $\|\omega_{i,j} \cap \omega_{u,v}\|$. In order to enable fast data exploration, we pre-compute a transition probability matrix and store it in an external file. For $n$ axes and $k$ uniform intervals, the total number of bins to be computed and stored is $\frac{(n-1)k^2}{2}$. Whenever a new Markov Chain is determined by the user, $n$ reads suffice to compute a joint probability value. In addition to the transition probability, we also pre-compute a bin map which stores the data frequency in each dimension, namely $\|\omega_{i,j}\|$, which requires $kn$ bins in total.

## 4. Visualization and Analysis

Once we have obtained a list of probability values for all data items from Equation (2), this can be represented either by a scatterplot or a histogram. A scatterplot can be used when rendering a small number of data items, whereas a histogram is adopted when rendering a large data set. In the first example, we would like to consider a synthetic dataset about the geometric features of pollen grains consisting of 3848 observations with 5 variables. This is the 1986 ASA Data Exposition data set from David Coleman of RCA Labs [WL97]. From the probability scatterplot shown in the third row of

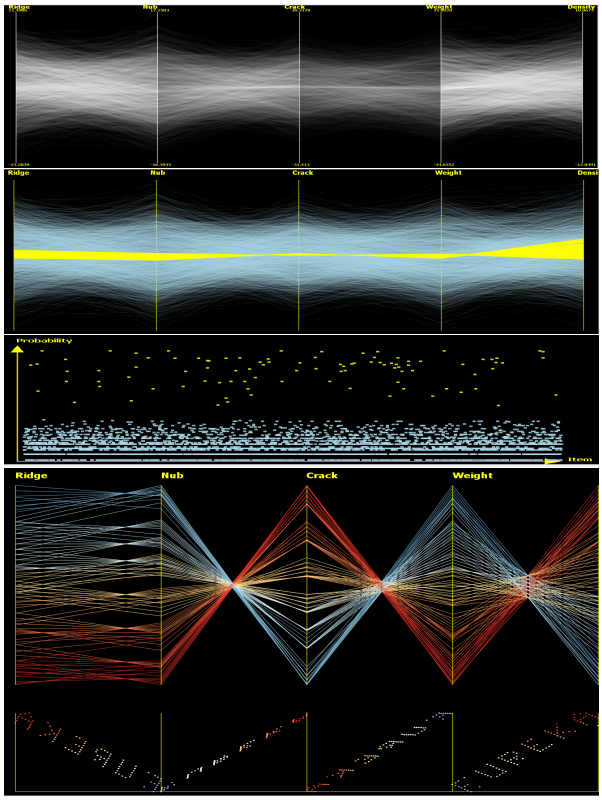


**Figure 3:** *This figure shows the pollen data set. The first row shows the line based histogram. The second row shows the composite brushing, with the yellow polylines showing the trend and the blue polylines the noise. The third row shows the scatterplot of probability distribution. The fourth row shows only the yellow points captured in the third row. A color scale is mapped to each polyline according to its position in the first axis of parallel coordinates.*

Figure 3, we can see that most of the data items have relatively low probability values as depicted in blue. However there are 99 data items having a much higher probability and are isolated from the original 3848 points as depicted in yellow. If we render these data trends and noise separately on parallel coordinates in different colors, we obtain the visualization on the second row in Figure 3. As we re-scale the selected data trends drawn in yellow, we are able to catch six clusters, as shown in the fourth row of Figure 3. Although the extracted 99 data points are only approximately 2.7% of the data set, we are able to succsessfully isolate these points from the noise. In the previous method [WL97], the time cost to prune the noisy data is approximately within three minutes. However, in our method the identification occurs instantly once we have computed the joint probability distribution. The first row of Figure 3 is a visualization using the line-based histogram [NH06]. The patterns in the middle of

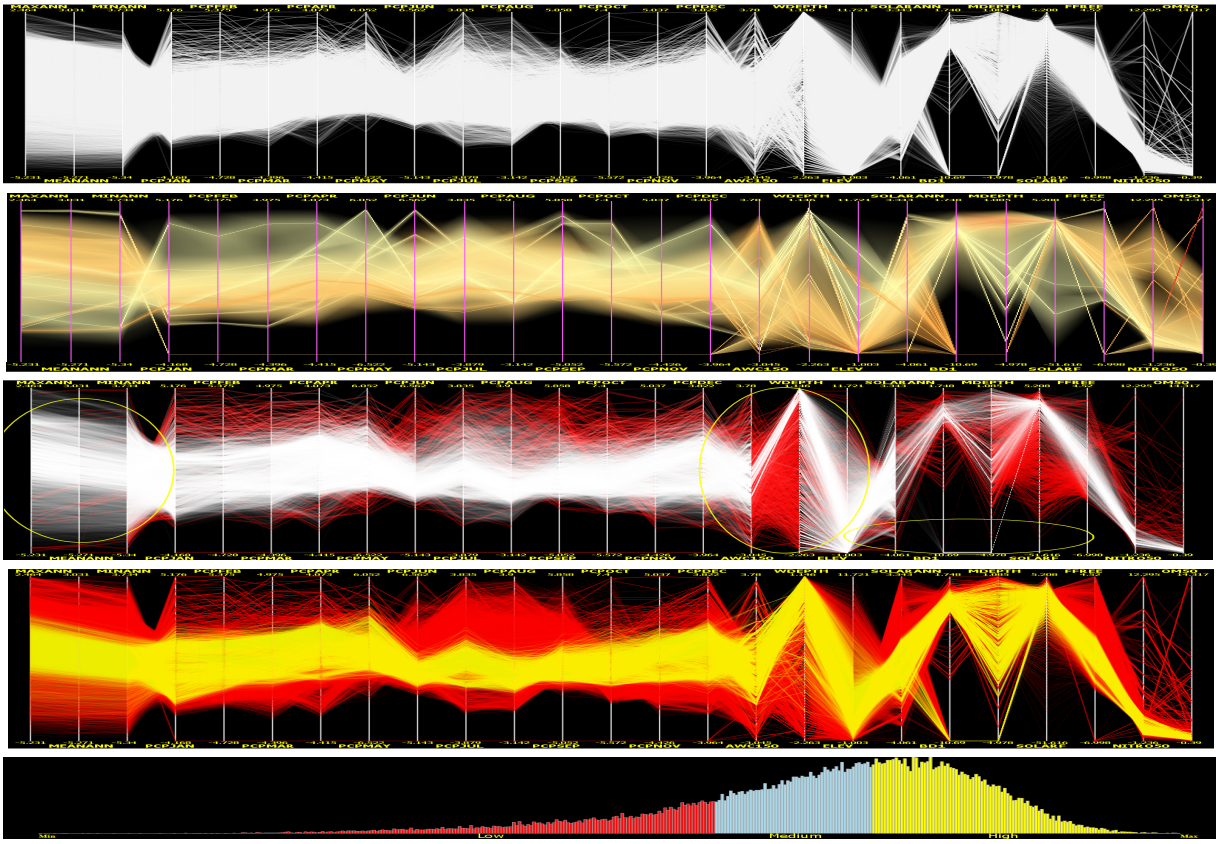

**Figure 4:** *This figure shows a proprietary, biodiversity data set from XMDV [XMD11]. This data set has 25 dimensions and 49324 samples. The first row is visualization using alpha blending. The second row shows the visualization rendered in hierarchical clustering and proximity-based representation. The third row shows the outlier-preserving line-based histogram [NH06]. The fourth row shows our composite brushing with yellow patterns representing a high probability range and red patterns a low probability range. The fifth row shows our probability histogram.*

the axes are not very sailent and clear. In addition, we are unable to extract and seperate these data points out of the noise from this method.

For a large data set, we could apply histogram for probability representation. Our goal is to select a subset of the data which can mostly represent multi-dimensional features and characteristics. In this section, we introduce some of the brushing techniques to handle this class of data sets. The essence of our technique is that a data sample in a multidimensional principal trend will have a high joint probability value, whereas a data item regarded as an outlier often has a very low probability value. The question arises as to how to classify the high, medium and low probability ranges. We provide a classification as guidance for the user to brush for data patterns. The user can also directly interact with different probability ranges by brushing. Initially, we compute the $10^{th}$ percentile, median and $90^{th}$ percentile values of the probability lists of all data samples. Then we use these values as the initial means for K-means clustering. The number of clusters namely $k$, is set to three as default referring to "high", "medium" and "low" probability range. In addition, we allow the user to choose an arbitrary number of clusters for their desired degree of classification. One point we emphasize is that the classification based on this probability distribution is used for data extraction and brushing, rather than data clustering.

In this example, we consider a real biodiversity informatics data set. This data set is from a clustering-based niche envelope model that William Hargrove and Forrest Hoffman studied for Lobolly pine across the contiguous United States [HH00]. From this data they aim to classify the Lobolly pine based on twenty-five factors, including elevation (ELEV), maximum, mean and minimum annual temperature (MAXANN, MEANANN, MINANN), monthly precipitation (PCPJAN to PCPDEC), several soil parameters, number of frost-free days (FFREE) and solar output and input. Each data element represents a data map which was developed for the continental United States at a resolution of 1

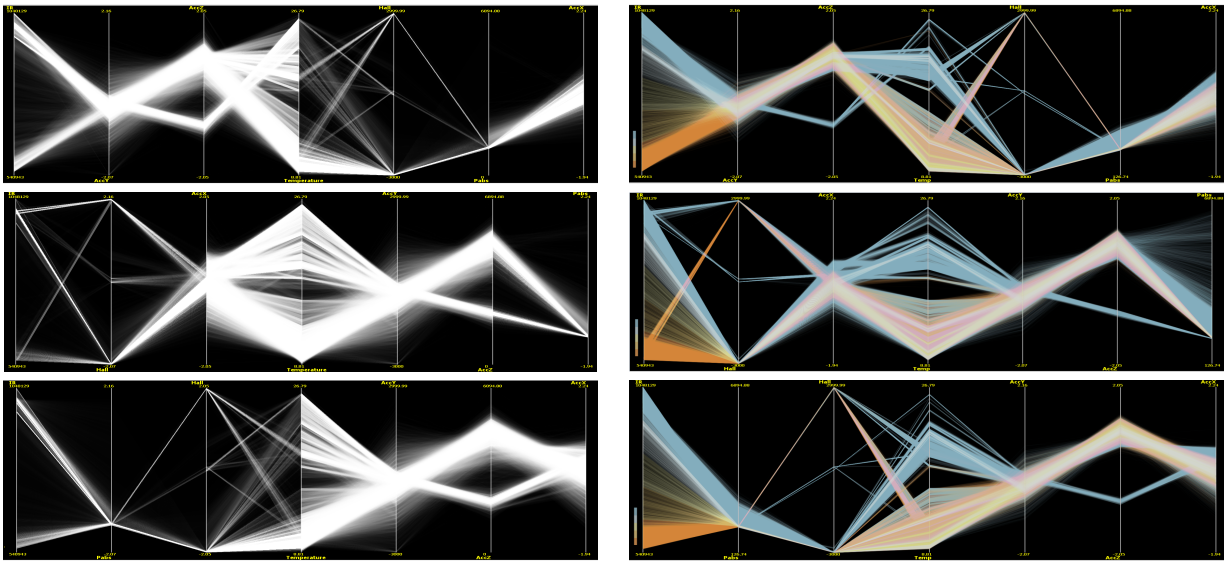

**Figure 5:** *This figure shows the visualizations of three different orderings of our animal tracking data set. The data patterns on the left column are rendered using a line-based histogram, while the patterns on the right column is the brushed data samples with high probability using our method. We note that a color scale is mapped to the position of polylines according to the first vertical axis in the parallel coordinates view to depict the coherent patterns in higher dimensions. In addition, we are unable to assign a color mapping on the n-dimensional data space on the patterns in right column, because they only show patterns in two-dimensional subspace.*

$km^2$. This data set has 49324 samples. It can be downloaded from the XMDV website [XMD11]. Shown in the bottom of Figure 4 is a histogram representation for the probability distribution. The histogram is partitioned into three parts using K-means clustering. Each class is depicted in different colors. If we brush the high probability cluster (depicted in yellow) and low probability cluster (depicted in red) respectively, we are able to obtain a visualization shown in the fourth row of Figure 4. The pattern in yellow illustrates a principal data trend. There are 45.5% of the data samples brushed which reveals a coherent pattern propagating through twenty five dimensions. In these areas, the overal annual temperature is moderate as shown in the first three axes. From axis ELEV we can see these areas are low in elevation. The precipitation has larger variance from January to May than from June to October. In addition, the precipition drops from May and remains relelvatively low in the next few months. The depth to water table (WDEPTH) reamins high.

## 5. Comparison

In this section, we would like to compare our results with other popular large data visualizations, such as alpha blending [Weg90, WL97], hierarchical clustering [FWR99] and line-based histograms [NH06, RK08]. The first row in Figure 4 is a visualization rendered using alpha blending. The density of the plots is represented with transparency. Under a low alpha value, the sparse parts of the dataset fade away while the more dense areas are emphasized. This works well with small datasets, however, with large datasets the range of data is much greater and consequently it is more difficult to fully represent the fidelity of complex datasets. It is difficult to obtain a clear understanding of patterns and clusters, as it becomes cluttered in some areas between axis AWC150 and ELEV. The yellow patterns in our method as shown in the fourth row of Figure 4 provides a much clearer data trend. In addition, outliers may get lost using alpha blending, such as patterns on the bottom between axis PCPJAN and PCPDEC. Our method is able to preserve such outliers as depicted in red patterns. If we combine the yellow and red patterns in our method, we are able to approximately reconstruct a complete view of data features with an emphasized view of principal data trends.

The second row of Figure 4 shows a hierarchical parallel coordinates rendered by XMDV [FWR99]. In this approach, a Birch's hierarchical clustering algorithm is adopted which builds a tree of nested clusters based on proximity information. Proximity-based coloring is introduced to demonstrate clusters, and transparency to show the mean and the extent of each cluster. Then multi-resolution views of the data can be rendered. Compared with the alpha blending, it is able to offer a clearer data distribution and preserve the low frequency data samples. However, the densities of different clusters might be difficult to distinguish from XMDV. This is because most of polylines are rendered in a small portion of screen space, the differences in the proximity-based

transparencies for various levels of clusters are not easy to discern. Using our method, we are able to immediately catch a clear view of central data trends as depicted in yellow and minor data trends depicted in red. In addition, because each polyline rendered is a mean value of a cluster in XMDV, they may deviate from the original positions of the polylines and cause problems with interpretation. For example, outliers on the bottom between the axis PCPJAN and PCPDEC are supposed to be at minimal value of each axis and patterns on the lower part between axes ELEV and SOLARANN are different from patterns in alpha blending and our method.

The third row of Figure 4 shows a visualization rendered using line-based histograms [NH06, RK08]. It is built upon a two-dimensional bin map storing the frequency of line segments in every neighboring dimensions, which is similar to our method. Eventually, every bin is rendered as a parallelogram connecting a pair of intervals at adjacent axes with its vertexes placed at the respective positions of the bin borders with its frequency represented by transparency. Then the high frequency histograms are emphasized whereas the low frequency ones fade away. Although the local data trends can be discovered from this method, the global trends in higher dimensions are missing. This causes discontinuity across high dimensional space, as highlighted in yellow. Our method is advantageous by overcoming such discontinuity by offering a coherent global data trend in multi-dimensional space. The previous method [NH06] also proposes an outlier detection method. For any low frequency bin, a 3 by 3 isolation filter is used to check the occupancy values of the 8 bins that are adjacent to the central bin. If the number of empty neighboring bins is above a certain threshold (say 2 for the corners, 4 for the borders and 6 or 7 for the rest), the central bin is declared an outlier which is shown as the red patterns in the third row of Figure 4. As we can see, the outliers obtained are limited to two-dimensional subspace which lose the continuity in higher dimensions. However, our method is able to present a continuous multi-dimensional outliers by brushing a low probability range.

## 6. Dimension Reordering

As our paper title implies, the focus of this paper is on the visual pattern analysis. Since reordering the parallel coordinates often leads to different visual patterns, therefore our probability distribution is optimized for these orderings in order to achieve the best visual effect. In section 3, we precompute a transition probability matrix and store it in an external file. Based on this matrix, computing a new probability value for each re-ordered data tuple is very fast, which requires $n$ additions where $n$ is the number of dimensions. Because our initiative is to extract the major patterns in visual space, therefore the change of individual element probability value in data space is of no interest in this paper. As long as the probability distribution is optimized, we can always display data samples with highest probability values to form the principal visual patterns in screen space to overcome visual clutter. This is the same as the previous output-oriented line-based histogram [NH06]. The difference with previous approach is that we consider the n-dimensional coherent visual patterns rather than two dimensional discontinuous visual patterns.

In this example, we consider a real world, large, marine biology data set [GJL*09]. Biologists at Swansea university have collected a large amount of data relating to animal movement by attaching sensors to individual subjects. The data here is re-sampled once a second over five days. In this example, we select 7 important data attributes with 536,548 records. This data set can be plotted using traditional parallel coordinates, but suffers from heavy overplotting, as shown in the top image of Figure 6. Shown in Figure 5 are the different orderings of our marine biology data set. Visualizations on the left column are rendered using the traditional line-based histogram and on the right column are the polylines brushed by the high-probability data items using our method. In our method, each polyline is assigned a different color along a user-defined axis, in our case the first axis (IR) is chosen. As we can see, no matter which ordering it is, the line-based histogram is able to illustrate local data trends in a two dimensional subspace, whereas our method manages to capture global data trends in higher dimensions. Then we are able to observe relationships and dependencies between any dimensions as rather than in neighboring dimensions in previous method.

## 7. Use Cases

In this section, we provide an in-depth analysis of the data trend discussed in Section 6 on our marine biology data set. As mentioned in the previous section, there are seven measurements in this marine biology data set, including Tri-axial accelerometer data in $x$, $y$ and $z$, mouth sensor(Hall), pressure, IR(speed) and temperature. As shown in the first row of right column in Figure 5, two major clusters along the first axis (IR) are revealed in different colors. If we decompose these two clusters by the first axis and render them separately, we are able to obtain the data patterns shown in Figure 6. The second row of Figure 6 shows the data trends with high IR value and the bottom row shows the trends with low IR value. The IR value indicates a relative speed of the animals moving against water, high IR often suggests a low speed and low IR indicates a high speed [LPL*08]. When the relative speed is low, as shown in the second row of Figure 6, there are two patterns can be observed with one leading to high temperature and the other to low temperature. Because the deeper the animal is swimming in the water, the lower the temperature. Combined with the pressure value (Pabs), we can infer two actions. One is that the animal is diving into the water and the other is ascent up to the surface. Moreover, from the mouth sensor (Hall), we can observe two patterns, one is that the animal closes its mouth when it is moving and the other is it opens mouth when preying. The data trends shown in the bottom of Figure 6 is a cluster with low

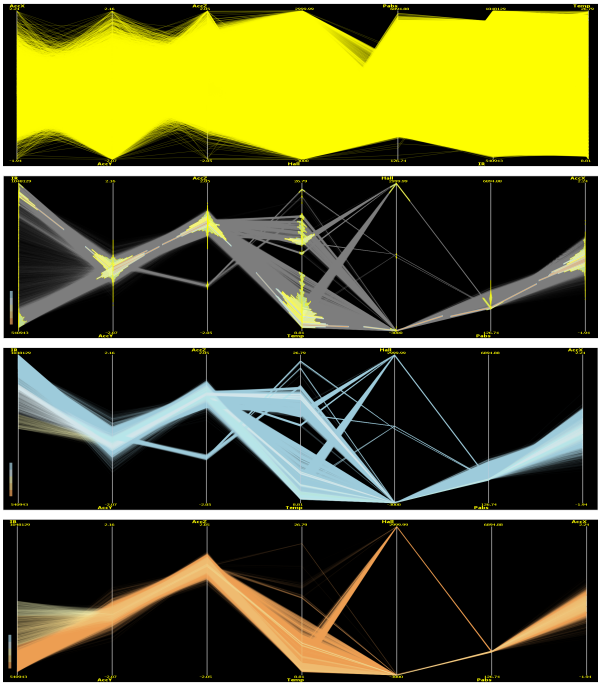

**Figure 6:** *The top row shows the angular histograms imposed on the brushed data trends in our marine biology data set. The middle and bottom row is the divided patterns by the first axis (IR) from the brushed polylines. A complete color-coded view is shown on the top of right column in Figure 5.*

IR value, which means that the animal is moving at a high speed. This pattern mostly leads to low temperature, from which we can infer that the animal is diving or swimming quickly under the water. Compared with the line-based histogram shown in the first row of left column in Figure 5, our method preserves salient global features of the data, uncovers different clusters and avoids pattern discontinuity across n-dimensions. By exploring the angular histograms, we are able to observe the data density in the brushed data patterns, especially in a large data set.

## 8. Conclusion and Future Work

In this paper, we have developed a Markov Chain model for visualizing multidimensional patterns with parallel coordinates. A histogram or scatterplot view presents the joint probability distributions for all data samples. The user is able to brush the data trend based on this probability value. Using our method, the global data trends in higher dimensional space can be discovered and displayed.

## References


[AdOL04] ARTERO A. O., DE OLIVEIRA M. C. F., LEVKOWITZ H.: Uncovering Clusters in Crowded Parallel Coordinates Visualizations. In *IEEE Information Visualization Conference* (2004), IEEE Computer Society, pp. 81–88. 1, 2

[BBP08] BLAAS J., BOTHA C. P., POST F. H.: Extensions of Parallel Coordinates for Interactive Exploration of Large Multi-Timepoint Data Sets. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1436–1451. 1, 2

[Car91] CARR D.: Looking at Large Data Sets Using Binned Data Plots. *Computing and Graphics in Statistics,ed. by Buja, A., Turkey,P.A* (1991), 7–39.

[FKLT10] FENG D., KWOCK L., LEE Y., TAYLOR R. M.: Matching Visual Saliency to Confidence in Plots of Uncertain Data. *IEEE Transaction on Visualization and Computer Graphics 16*, 6 (2010), 980–989. 2

[FWR99] FUA Y.-H., WARD M. O., RUNDENSTEINER E. A.: Hierarchical Parallel Coordinates for Exploration of Large Datasets. In *IEEE Visualization* (1999), pp. 43–50. 2, 6

[GJL*09] GRUNDY E., JONES M. W., LARAMEE R. S., WILSON R. P., SHEPARD E. L. C.: Visualisation of Sensor Data from Animal Movement. *Computer Graphics Forum 28*, 3 (2009), 815–822. 7

[GPL*11] GENG Z., PENG Z., LARAMEE R. S., WALKER R., ROBERTS J.: Angular Histograms: Frequency Based Visualizations For Large, High-Dimensional Data. *IEEE Transaction on Visualization and Computer Graphics 17*, 6 (2011), 2572 – 2580.

[HH00] HARGROVE W. W., HOFFMAN F. M.: An Analytical Assessment Tool for Predicting Changes in a Species Distribution Map Following Changes in Environmental Conditions. In *4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4): Problems, Prospects and Research Needs* (2000), pp. 11 – 18. 5

[ID90] INSELBERG A., DIMSDALE B.: Parallel Coordinates: A Tool for Visualizing Multi-dimensional Geometry. In *Proceedings of IEEE Visualization* (1990), pp. 361–378. 1

[LPL*08] L.C.SHEPARD E., P.WILSON R., LIEBSCH N., QUINTANA F., LAICH A. G., LUCKE K.: Flexible paddle sheds new light on speed: a novel method for the remote measurement of swim speed in aquatic animals. *Endang Species Res 4*, 6 (2008), 157–164. 7

[NH06] NOVOTNY M., HAUSER H.: Outlier-Preserving Focus+Context Visualization in Parallel Coordinates. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 893–900. 1, 2, 4, 5, 6, 7

[Rab89] RABINER L. R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE 77*, 2 (1989), 257–286. 2, 3

[RK08] RUEBEL O., K. W.: High Performance Multivariate Visual Data Exploration for Extremely Large Data. Lawrence Berkeley National Laboratory, Barkeley, CA 94720, USA, 2008. 2, 6, 7

[RTT03] RODRIGUES J. F., TRAINA A. J. M., TRAINA C.: Frequency Plot and Relevance Plot to Enhance Visual Data Exploration. In *SIBGRAPI* (2003), IEEE Computer Society, pp. 117–124.

[Weg90] WEGMAN E. J.: Hyperdimensional Data Analysis Using Parallel Coordinates. *Journal of the American Statistical Association 85*, 411 (1990), 664–672. 6

[WL97] WEGMAN E. J., LUO Q.: High Dimensional Clustering Using Parallel Coordinates and the Grand Tour. *Computing Science and Statistics 28* (1997), 352–360. 4, 6

[XMD11] XMDV: Data Sets, 2011. `http://davis.wpi.edu/xmdv/index.html`. 5, 6