Pacific Graphics 2015
N. J. Mitra, J. Stam, and K. Xu
(Guest Editors)

Volume 34 (2015), Number 7

EasyXplorer: A Flexible Visual Exploration Approach for Multivariate Spatial Data

Feiran Wu[1] Guoning Chen[2] Jin Huang[1] Yubo Tao[1] Wei Chen[†1]

[1]State Key Lab of CAD&CG, Zhejiang University
[2]Department of Computer Science, University of Houston

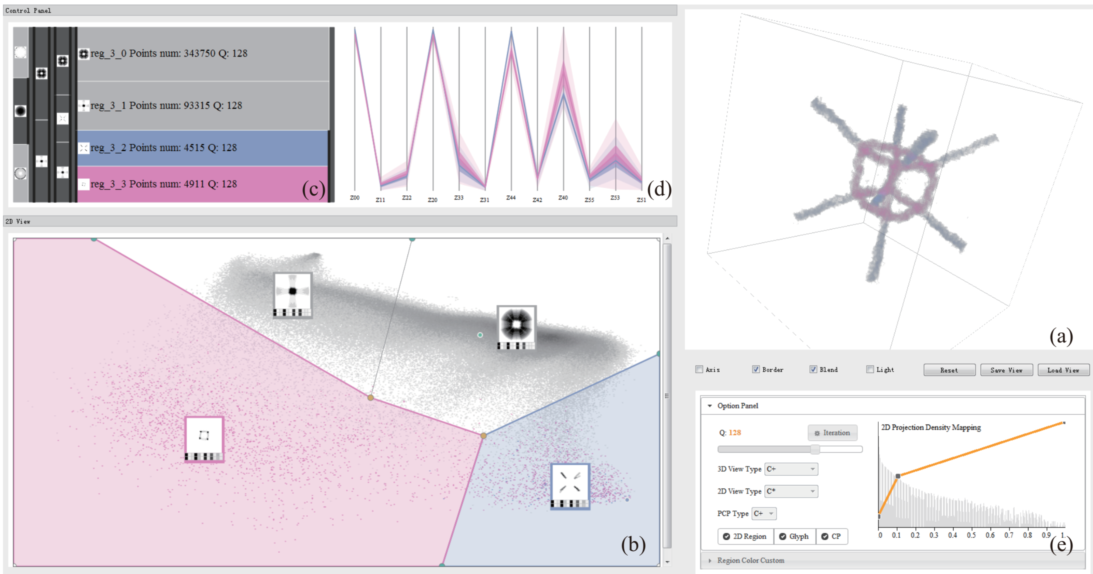

Figure 1: *The interface of EasyXplorer with the cubic symmetry field as an example. (a) The 3D view shows the classified result. (b) The 2D view visualizes the projections of all data points and the classification corresponding to (a). The statistics of multiple variates of each region are encoded with a set of glyphs. A color-filled 2D Voronoi graph is used to augment the navigation and manipulation of the clusters. (c) The flow chart for recording the steps of exploration. (d) The parallel coordinates view for comparing among different regions in the same category. (e) The controlling widgets for adjusting the visualization parameters.*

Abstract

Exploring multivariate spatial data attracts much attention in the visualization community. The main challenge lies in that automatic analysis techniques is insufficient in discovering complicated patterns with the perspective of human beings, while visualization techniques are incapable of accurately identifying the features of interest. This paper addresses this contradiction by enhancing automatic analysis techniques with human intelligence in an iterative visual exploration process. The integrated system, called EasyXplorer, provides a suite of intuitive clustering, dimension reduction, visual encoding and filtering widgets within 2D and 3D views, allowing an inexperienced user to visually explore and reason undiscovered features with several simple interactions. Case studies show the quality and scalability of our approach in quite challenging examples.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications—Multivariate 3D Data; Visual Analysis

† Corresponding Author:chenwei@cad.zju.edu.cn

1. Introduction

Multivariate spatial data refers to the data that is defined in 3D space and contains multiple independent or dependent variables at each data point. Analyzing multivariate spatial data is of great importance in many science and engineering applications like medical imaging, climate research and Computational Fluid Dynamics. Yet, spatial classification and feature identification of multivariate spatial data remains an open problem in visualization community, mainly due to the unknown feature patterns and the lack of prior knowledge about the data distribution. The difficulty is aggravated by the fact that there are multiple variables whose relations are subtle, latent and complicated.

Conventional visualization techniques for 3D scalar fields can show the structures formed by the scalar values of the field. However, they can only display one variable at a time for multivariate spatial data. 3D vector fields can be regarded as special multivariate spatial data that represents the velocity and geometric information. Approaches for visualizing them either generate geometry structures or glyphs [PVH*03] to characterize features of interest, or employ dense texture to depict the patterns [LHD*04]. These techniques have proven to be very effective for depicting the appearance, but are hardly capable of handling general multivariate 3D data, e.g., symmetric tensor field data [KASH13].

Much attention has been paid to the visualization of multivariate non-spatial databy abstracting the physical implications of data attributes. Typical solutions include dimension reduction [KSC*10], iconography, density-based display [ZBDS12], and scatterplot matrix.Recently, there is a trend to integrate these approaches into the exploration of the spatial data. Most of them seek to address the problem of multi-dimensional transfer function design for visualizing 3D scalar fields [WZK12]. By combining dimension reduction and parallel coordinates techniques, the dimension relations in multivariate spatial data can be progressively disclosed [GXY12]. However, existing solutions largely rely on the user to explore each dimension and investigate their relations. For a novice user, the exploration process can be counter-intuitive and laborious, and may require a long learning process.

The gap between the flexibility of multivariate data visualization and the fidelity requirement of spatial data exploration makes the visual classification and feature identification of multivariate spatial data quite troublesome. We have identified three reasons. First, the feature search space is too large that it costs the user much time on understanding the underlying features and their spatial relationships. Second, modulating the parameters of multi-dimensional visualization and classification widgets to maximize the likelihood of feature separation is a non-trivial task. Meanwhile, the mapping from multivariate spatial data to visual components is much more difficult than for 3D scalar fields. Third, regions of interest (ROIs) in multivariate spatial data are distributed irregularly in the spatial domain. Distinguishing them from other data parts requires a sequence of careful yet laborious operations.

The main contribution of this paper is the systematic description of integrating different visualization and analysis techniques and its application to the exploration of multivariate spatial data. We enhance conventional multivariate spatial data visualization schemes by integrating a suite of clustering, dimension reduction, interaction, filtering and visual encoding techniques within a 2D/3D dual visual interface. By decomposing the analysis process into a clustering-projection-classification iteration, the user is empowered with a scalable explorer for the inspection of correlations among different variables in the higher dimensional space in a coarse-to-fine fashion. The integrated system, EasyXplorer, provides an intuitive visual exploration and a reasoning tool that assists the user in identifying, locating, distinguishing, categorizing, comparing, associating, or correlating the underlying data. The case studies on several challenging datasets demonstrate that our approach compares favorably with conventional methods in both the scalability and the efficiency.

2. Related Work

2.1. Visual Exploration of Multivariate Spatial Data

Existing multivariate spatial visualization approaches generally employ the techniques developed for non-spatial data [KH13]. Multiple linked views, dimension reduction and parallel coordinates are among the most popular techniques. The first one visualizes the dataset from multiple aspects within a connected visual interface [GRW*00] [Dol07]. Dimension reduction is a standard scheme for high-dimensional data analysis by projecting a high-dimensional point set into a low-dimensional space. Representative techniques include the local linear embedding [ZK10], multi-dimensional scaling [GXY12] and other methods [JBS08]. Parallel coordinates technique also attracts much attention because it allows for showing and manipulating the individual variables or dimensions at the same time [TPM05] [KERC09] [ZTM*13]. Besides the specific techniques, we also inspired by the idea that combining the processing power of the computer with the capabilities of the human user [FWG09].

2.2. Visualization of Multivariate Spatial Data

In general, glyph and texture are fundamental means for encoding important information in multivariate data visualization. The glyph representation is widely used in vector and tensor field visualization [RP08],and general multivariate spatial datasets.Typically, texture is used together with the color [UIM*03] for depicting multivariate information.There are a large body of techniques for visualizing 3D vector and tensor fields. The cross-frame fields used in case study are certain rotational symmetry fields. A 2D cross-frame field can be visualized with the line integral convolution (LIC) technique or other well-designed second-order tensor field visualization approaches [PZ11] [HTWB11] [HPC*13].To our best knowledge, until now there is not an effective method to visualize a 3D cross-frame field due to the inherent ambiguities of the cross-frame field.

2.3. Visual Classification of Multivariate Spatial Data

The clustering of spatial data has received much attention in the past decade. A common way is to convert multivariate spatial data into a statistical space, or compute a set of statistical variables, and explore and analyze the underlying data in the statistical space [HPB*10]. In volume visualization, this problem is commonly known as volume classification. The representative scheme is the multi-dimensional transfer function design which enables the user to manipulate a multi-dimensional histogram of derived attributes [RBS05] [LYL*06], a dimension reduction representation [KSC*10], or an attribute space with an information metric [MJW*13] to explore the 3D regions. Interactive feature extraction is the other way to classify the target from the time-varying flow simulations data [MM09] and vector field [DAN*10]. etc. Wang et al. [WZK12] introduce a modified dendrogram to represent the feature space clusters. This cluster-and-analyze scheme is also adopted and augmented in EasyXplorer by providing a comprehensive and flexible visual interface. Further, the difference between these methods and our approach is the iterative analysis process which reuses the cluster-and-analyze scheme time after time to satisfy the requirement of exploration. The detail will be illustrated in the next section.

3. Our approach

Let $V = \{\mathbf{v}_n, n = 1,2,...,N\}$ be a multivariate spatial dataset with N data points, and $P = \{\mathbf{p}_n \in R^3, n = 1,2,...,N\}$ be its associated physical positions. An ROI is typically continuous in the physical space, and the distribution of its variates is concentrated in a region within the attribute space. A straightforward way is to define a distance metric concerning the variates of data points, and classify the entire dataset into multiple regions by means of a 3D clustering process. However, we conclude four problems based on this scheme:

P1 ROI evaluation There is no sufficient and objective standard to justify whether ROIs are accurately computed.

P2 ROI refinement The ROIs derived by an automatic algorithm also contain redundancy or deficiency and need to be refined. However, direct manipulation of the ROIs in spatial space is troublesome.

P3 Parameter adaptation A special set of parameters may create a pleasing result for a dataset. Nonetheless, it is intractable to create desired results for various datasets with a uniform set of parameters.

P4 Generality Besides some common data field, for some datasets (e.g. 3D cross-frame fields), it is still troublesome to explore the clustering results with conventional 3D visualization techniques.

In general, EasyXplorer addresses these challenges by integrating various techniques into an iterative analysis loop. In this loop, the unconcerned parts of the underlying dataset are iteratively culled by means of parameter modulation until extracting the ROIs. In each iteration, a spatial clustering operation is firstly performed to classify the entire dataset into multiple regions as the candidates of ROIs, which is equivalent to computing the optimal partition $C = \{C_i, i = 1,2,..,M, M \ll N\}$, of which C_i contains N_i data points in V, and has a distinctive variate distribution from others in C. Then the user refines and analyzes these regions by incorporating the user expertise and experience in an intuitive visual interface. The user can decide which region could be abandoned while the user-concerned regions will be selected as the input for the next iteration.

In particular, the solutions for the corresponding problems mentioned above are:

S1 and S2 EasyXplorer addresses P1 and P2 with a 3D-2D correlation interface. The data points are depicted in spatial space(3D) and attribute space(2D), respectively. The 2D/3D dual visual interface with the embedded visual encoding scheme will help the user evaluate the targeted data points by multi-perspective. In the terms of refinement, because the operation on 2D is easier than 3D, the 2D view may provide a flexible interface to refine the targeted data points. Sections 3.2 to 3.5 describe how to evaluate and refine by visual design and interaction.

S3 Some automatic clustering methods introduce several parameters to control the coarseness of the clustering [NN04].Instead of setting parameters blindly, the analysis iterative loop enables the user to make a coarse-to-fine exploration. The details are presented in section 3.5.2.

S4 To explore different types of data, EasyXplorer firstly pre-processes the data and converts them into a multiple-scalar dataset.

Figure 2 demonstrates the system pipeline. Below we discuss each step in detail.

3.1. Data Preprocessing

The data preprocessing is the initial step of the exploration. It varies for different types of datasets. In general, multiple variables of each data point can be regarded as the local feature description of the underlying dataset. For vector fields, tensor fields and some special fields, a specific local feature description is needed to characterize the local distributions of multiple variables and to remove the relevances among

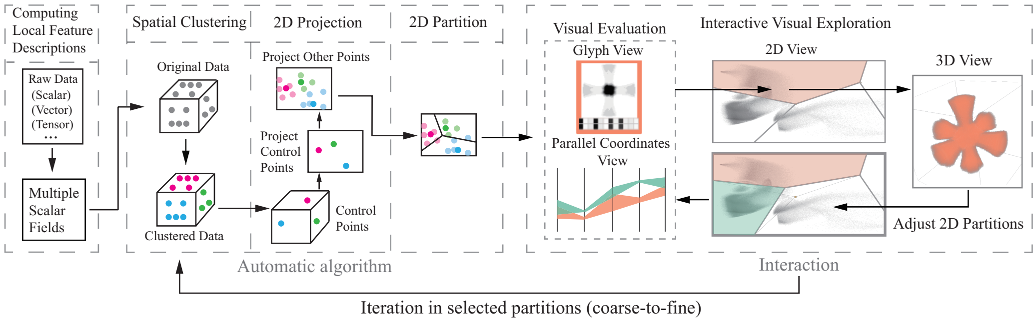

Figure 2: *The pipeline of EasyXplorer.*

variables. We will introduce the descriptions employed in 3D cross-frame fields as an example.

Computing the local feature description of a multivariate dataset yields another multivariate spatial field, in which each data point has multiple scalar attributes. The space spanned by these scalar attributes is denoted as the attribute space. For the sake of clarity, we resample the data to a regular grid and assume that each data point of the underlying multivariate dataset has a list of scalar attributes. Extending our approach to other cases is trivial by regarding each component of a vector or a tensor as a scalar.

3.2. Spatial Clustering

Here we employ the statistical region merging (SRM) algorithm [NN04] which iteratively merges regions by considering the proximity of the statistical information of the local feature descriptions. The coarseness of the clustering is monitored by a parameter Q which determines the granularity of the clustered regions. A large value of Q generates a fine region clustering result, and vice versa. In each iteration of the exploration process, Q is dynamically modulated under the steering of the analyst. In section 3.5.2 we will introduce how to adaptively set Q. We denote the clustered regions in spatial space as $C^* = \{C_i^*, i = 1,2,..,M^*\}$.

3.3. 2D Projection

According to C^*, EasyXplorer performs a low-dimensional (2D) embedding of all data points with respect to their proximities in the attribute space. In EasyXplorer, the 2D projection method has two functions: 1) Providing an attribute space view for millions or more data points; 2) Reflecting the proximity among C_i^*. Due to the large capacity of V, it would be intractable to use a conventional dimension reduction method (e.g., multidimensional scaling). Local affine multidimensional projection (LAMP) technique [JCC*11] is a decent technique that is capable of handling large-scale data and meets the first requirement. It projects a set of selected control points, and employs an affine transform to embed all data points based on the 2D positions of the control points.

To meet the second requirement, the control points are determined according to the clusters C^*. Let a sequence of control points be $CP = \{\mathbf{cp}_1, \mathbf{cp}_2, ..., \mathbf{cp}_{M^*}\}$ based on C^*:

$$\mathbf{cp}_i = \{\overline{\mathbf{v}}_i, w\overline{\mathbf{p}}_i\}, \quad i = 1,2,...,M^* \tag{1}$$

where M^* is the size of the C^*, w is an adjustable weighting parameter. $\overline{\mathbf{p}}_i$ is the centroid of the physical positions of all data points in C_i^*, and $\overline{\mathbf{v}}_i$ denotes the mean value of the multivariate of all points in C_i^*.Both the 3D position $\overline{\mathbf{p}}_i$ and associated attributes $\overline{\mathbf{v}}_i$ of a control point $\mathbf{cp}_i$ are used to compute the proximity d_{ij} among all control points as $d_{ij} = \|\mathbf{cp_i} - \mathbf{cp_j}\|_2$.

Thereafter, all control points are embedded into the 2D space by means of the standard multidimensional scaling algorithm. Subsequently, all data points are projected by means of the LAMP algorithm. Substantially, w controls the influence of spatial position on the distribution of 2D projection. In practice, we set $w = 0.1$ to drive the 2D projection led by the attribute space.

3.4. 2D Partition

To refine C^*, this step generates the counterparts of C^* in the attribute space and formats them as a user-adjustable structure. We denote them as $C^+ = \{C_i^+, i = 1,2,..,M^+, M^+ = M^*\}$, which is the partition in 2D space. Within the 2D projection, the Euclidean distance in the 2D space approximately describes the proximity among the control points and the data points. Thus, the data points belonging to C_i^* may distribute around its control point in the 2D space.

EasyXplorer employs the Voronoi graph to preset the partition of the entire 2D space, where the control points are considered as the Voronoi sites. The edges of each Voronoi cell partition the 2D projection and form the new regions C^+. Because points of a Voronoi cell tend to be closer to their Voronoi site (control point), the Voronoi graph can obtain the reasonable partitions based on the 2D projection and match the C_i^+ to each corresponding C_i^*. In Figure 8 (a), the Voronoi graph divides the 2D projection into 3 partitions.

The color indicates the point distribution of different clusters. Note that the colored 2D projection highlights the distribution of the points belonging to each C_i^*. After that, the user can interactively modify the boundary of C^+ for further exploration.

3.5. Interactive Visual Exploration

We design a series of views to integrate the decision of the user into our analysis loop. First, we visualize the C^* and C^+ in 3D view and 2D view, which substantially support to depict the distribution pattern in the physical space and the corresponding attribute space. Then, we encode the relevant information by the glyph view and the parallel coordinates view, which provide a user-friendly interface to support the user to evaluate each C_i^* with its corresponding C_i^+ or compare the subsets in C^* or C^+. Furthermore, a flow chart is adopted to record and trace the whole analysis process. For the convenience of illustration, we denote a C_i^* with its associated C_i^+ as an associated pair (C_i^*, C_i^+) below.

3.5.1. Interface

3D and 2D views To illustrate a selected associated pair (C_i^*, C_i^+), by default, the 3D view visualizes the spatial distribution of C_i^+ by volume rendering (Figure 1 (a)) while the 2D view shows the projection distribution of the corresponding C_i^* by default (Figure 1 (b)). The user can flexibly switch between C_i^* and C_i^+ in these two views to observe the (C_i^*, C_i^+) in spatial space or attribute space.

In 3D view, an index volume for the switched C^* or C^+ is built. Each voxel in the index volume has one corresponding data point in the multivariate dataset. The value of each voxel is defined as the scaled index:

$$s_i = i\frac{S}{M+1}, i = (1, 2, ..., M) \tag{2}$$

where i is the index number of voxel's corresponding C_i^* or C_i^+, s_i is the scaled index, and S is the range of the voxels in the index volume (255 in our implementation). A 1D transfer function is adopted to assign colors to each partitioned region. We simplify the transfer function and the user can intuitively select the color and opacity of partitioned regions to either highlight or hide them. It should be noted that (C_i^*, C_i^+) share the same color assigned by color custom panel in all the views.

The 2D view firstly shows the 2D distribution of data points after the 2D projection and the preset partition. The projection density is simply accumulated, yielding a heatmap visualization. The density from low to high is mapped to grey with the decreased lightness, which is controlled by a 1D transfer function in the option panel (Figure 1 (e)). The preset Voronoi partitioned regions are bounded by polygons. The circles in different colors indicate the locations of the sites of Voronoi regions (also the control points of the 2D projection). The vertices of these polygons serve as the anchors. The user can drag these anchors to modify the boundary of the corresponding regions.

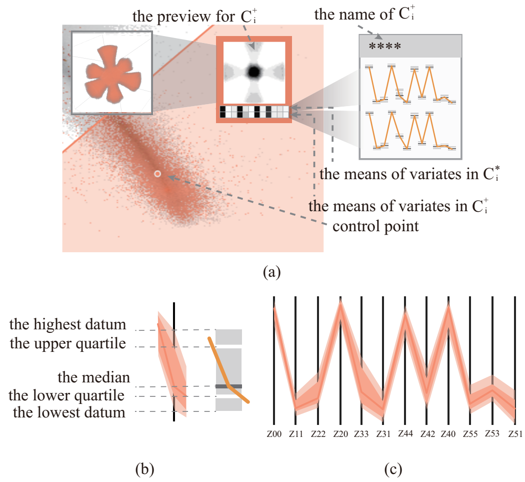

Figure 3: *(a) The constitution of the glyph view. (b) The statistic encoding scheme in the parallel set and the box-plot. (c) The parallel coordinates plot in C^+ mode. These views are visually connected by the same assigned color.*

Glyph View This view mainly shows the statistical information within (C_i^*, C_i^+). Each glyph view embedded in 2D view is located at the centroid of the corresponding partition C_i^+ and consists of several components.

First, the spatial distribution information is encoded by a snapshot as a preview for each partitioned region (Figure 3 (a)). For each C_i^+, the number of data points belonging to a C_i^+ along the Z direction in physical space is accumulated for each pixel in the X-Y imaging plane. Then the density distribution is visually encoded by gray scale color coding.

Second, the glyph view encodes the statistical information of C_i^* and C_i^+ in pixel style or box-plot style. Within the pixel style, two rows of pixels represent the mean of the variable in C_i^* and the corresponding C_i^+, respectively, which provides a simple glance of statistic within (C_i^*, C_i^+). For each row, every pixel from left to right represents a variable. Grey colors from light to dark encode the value from low to high (Figure 3 (a)). When the mouse hovers on the glyph, the view shows the name of the partition and the information in box-plot style (the right of Figure 3 (a)). The box-plot style shares the same order of the variables with the pixel style, and the upper and the bottom box-plot respectively represent variables in C_i^* and C_i^+. We demonstrate this detailed encoding scheme of the box-plot at the right of Figure 3 (b). Specially, we link each median of the box-plots by the orange polylines to highlight them.

Parallel Coordinates View Different from the glyph view, the parallel coordinates plot (PCP) offers the comparison among different subsets in the same category (e.g., C_i^+s in

C^+ or C_i^*s in C^*). Each axis represents one attribute with its name at the bottom. Instead of simply drawing the polylines onto the axes, we integrate the box-plot to the traditional PCP. When a C_i^+ or C_i^* is selected, the lower quartile, the upper quartile, the median, the lowest datum (within 1.5 interquartile range (IQR) of the lower quartile) and the highest datum (within 1.5 IQR of the upper quartile) of each attribute are encoded by the hybrid ribbon (Figure 3 (c)) in the color assigned for the associated pair (C_i^*, C_i^+). Figure 3 (b) depicts the detail of the visual scheme on an axis and compare it with the box-plot on the glyph view.

Flow Chart To support the iterative analyzing loop and remind the user of the history of analysis, we design the flow chart to record the whole process of iterative exploration(Figure 4). The partitioned regions in each iteration occupy the chart cells with a vertical layout. The information, such as the name, the preview, the points number of this region and the parameter setting in this iteration are listed in this cell. The height of each chart cell encodes the number of points that the corresponding region contains.When the next iteration is turned on, the history iterations are aggregated, showing only the preview image in the cell. The selected history regions are encoded by dark grey color while the selected regions in current iteration are shown in the assigned color. According to the colored cell, the user can figure out which partitioned regions join in each iteration step and query them by selecting the corresponding cell.

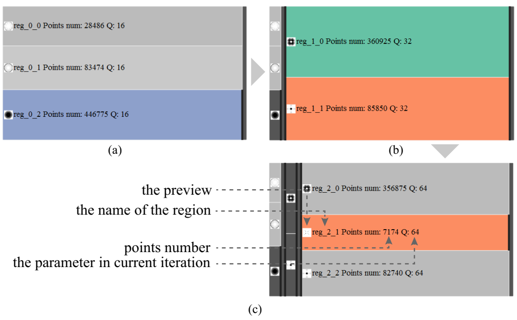

Figure 4: *The evolution of a flow chart within three iterations. (a) The first iteration extracts three partitions. (b) The user selects the third partition and executes the second iteration based on the data points belonging to this partition. The selected partition is highlighted by dark grey as the history selection marker. (c) The third iteration and the information introduction of the flow chart cell. The second partition has been selected and highlighted by the assigned color.*

Controlling Widgets The control panel contains a slide to modulate the granularity parameter Q, a density mapping curve as the 1D transfer function to adjust the display of the 2D projection, and a set of visual mode selectors for each view (Figure 1 (e)). We also provide the interface to select and modify the color and opacity of each associated pair.

3.5.2. The User Interaction

The initial interaction step of each iteration is modulating the parameter to generate the reasonable preset of 3D clusters. However, depicting too many clusters may cause visual clutter and user interactions. Besides, to narrow the feature search space accurately, we should ensure the points of ROIs would not be lost.

As a result, the parameter modulating in our system follows the rule called "coarse-to-fine" to generate a few of clusters in each iteration and abandon the non-interested points gradually. In the initial several iterations, the coarse clustering effectively reduce the number of clusters. The user then executes the later interactions to conservatively preserves the most valuable points. Because of the data pruning in the previous iteration, the finer clustering in the later iteration will not generate too many clusters. In our cases, this rule is implemented by assigning the Q value from low to high. Combining the practical experience with the parameter setting tactics in [NN04], we double the Q value in each iteration. Furthermore, The initial Q is also an empirical value, which is depended on the number of the clusters generated by this Q. In our cases, we set the initial clusters to be no more than 5 to avoid the visual confusion and simplify the analysis.

After that, the exploration in one iteration may follow the steps as 1) glancing at the 2D projection and the snapshots to decide which region to select, 2) labeling the interested regions, 3) evaluating the regions in physical and attribute space by the views and 4) refining the target regions by dragging the anchors of the 2D partition. The detailed operation will be illustrated in the case study.

4. Implementation

The main frame of EasyXplorer is implemented with Qt. To support the interaction in real-time, the computation-intensive tasks are all written in C++. The 3D visualization algorithm such as volume rendering is written with OpenGL. Besides, we select D3.js to draw 2D visualization widget. In practice, the automatic algorithm at the beginning of an analysis iteration spends most of time (depending on the data size, approximately 4 $\sim$ 8s in our cases). After that, the response of the interaction can be completed in real time.

5. Case Studies

5.1. 3D Nuclear Fusion Simulation Dataset

The 3D nuclear fusion simulation dataset records the frames within the simulation of the intermixing process between different fluid substances to observe the instability of a fluid interface. Five variables are incorporated in the dataset, including two scalar variables: density (D), temperature (T); a triple vector: velocity (the value of ρ, θ and φ in spherical coordinates are named as V0, V1 and V2 respectively).

We choose the 60th, 216th, 314th and 398th frame to analyze the time-varying pattern. The resolution of each frame is $128 \times 128 \times 128$. Because of its simple structure and few variables, we use this dataset as an instruction to illustrate the system operation. According to the knowledge from our domain experts, the boundary layer of the two fluid substances will be more and more active over time. The task is to study each frame successively and explore the change of fluid interface within the intermixing process. Below we describe the analysis process in detail by using the 314th frame as an example.

In the initial iteration, the granularity parameter Q is set to be 8, yielding 5 clusters and associated control points. All data points are projected to the 2D space, as shown in Figure 5 (a). A 2D Voronoi graph partitions the 2D data points into 5 regions. By seeing the 2D view, the user discovers the outliers according to the 2D projection (marked by the red circle in Figure 5 (a)). By selecting the preset regions, the user may compare the structure by 2D partition with 3D cluster. In the 3D view(Figure 5 (c)), the 3D clustering generates the coarse structure, while the 2D partition gives an incoherent (The region in pink is separated by the blue region, because of the inaccurate preset) but more smooth structure (Figure 5 (d)). The user then locates the fluid interface and unrelated structures by the following two observations: First, these two schemes all extract the outliers as the the core (the heat spot, one substance) and the outside of the fluid interface (the other substance). By the PCP view, the points in these regions are distinct in density and temperature (Figure 5 (a)). Second, the glyph view and the PCP view also show the value of the other regions is dispersed in each attribute (Figure 5 (a)), which may suggest that the instability exists at these regions. Because we know the fluid interface is active, the region in orange and blue is more likely be the target. At last, we smooth the coarse structure in the attribute space by adjusting the anchors (Figure 5 (b)(e)).

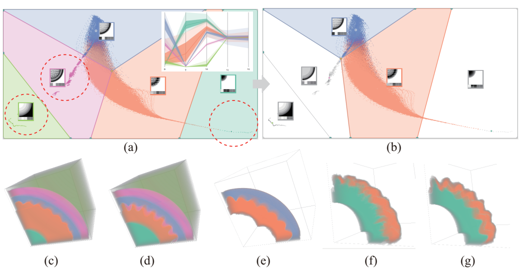

Figure 5: *(a) The result after the initial clustering. (b) After modifying the anchors, the user refines and preserves the regions in orange and blue. (c) The structure extracted by the spatial clustering in the first iteration. (d) The structure extracted by 2D partition based on the clusters in (c). (e) The 3D view of the selected regions in (b), which will be imported into next iteration. (f) The clustered structure in the third iteration. (g) The refined result of (f) by 2D partition.*

To disclose more details at the refined regions, the user removes the irrelevant points (the core and the outside of the fluid interface) and increases the granularity parameter Q. Subsequently, following the similar steps in the previous iteration, in the third iteration, we obtain a better fluid interface as the cyan structure depicted in Figure 5 (g). The user may observe the difference between the 3D clustering (Figure 5 (f)) and the refined result by 2D partition (Figure 5 (g)) by switching to the 3D view display mode and visualizing the selected regions. Obviously, the refined structure is more smooth and distinct. Figure 6 characterizes this time-varying perturbance structure in the selected time frames.

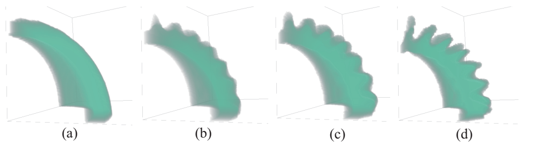

Figure 6: *The comparisons of the regions disclosed the 60th (a), 216th (b), 314th (c) and 398th (d) frames, respectively.*

5.2. Rotational Symmetry Vector Fields

The rotational symmetry vector field is one type of tensor fields, and is of paramount importance in many applications, such as 2D quadrilateral remeshing, 3D hexahedral remeshing, texture synthesis, and non-photo realistic rendering. A simple example is the N-rotational symmetry vector field (N-RoSy field) [PZ07] on a 2D manifold, of which each point has N unit vectors, and the angles between two neighboring ones are identical. In 3D, despite many computational and topological approaches,it still lacks of an effective means to visually analyze them. For instance, a point of a 3D cross-frame field contains six unit vectors that form a cubic symmetry [HTWB11], posing challenges for visualization.

Computing Feature Descriptions Singularities are the most important features of 2D/3D N-RoSy fields, and they are rotational invariant, namely, an arbitrary global rotation of the field does not change the distribution of them in the local frame. Accordingly, we construct a rotational invariant local feature description based on Zernike decomposition, which has been successfully applied to shape retrieval [KH90] [NK03].

Suppose that each point $\mathbf{p}$ in the underlying symmetry vector field has N unit vectors $\vec{r}_i, i = 1, \cdots, N$. In the spherical neighborhood $S(\mathbf{c})$ of a point $\mathbf{c}$, a scalar field $\rho(\mathbf{p}), \mathbf{p} \in S(\mathbf{c})$ is derived using Equation 3:

$$\rho(\mathbf{p}) = \max_i \left\{ \frac{\mathbf{p} - \mathbf{c}}{\|\mathbf{p} - \mathbf{c}\|} \cdot \vec{r}_i \right\}. \tag{3}$$

The above equation is equivalent to finding the vector that best matches the radial direction $\mathbf{p} - \mathbf{c}$, and using its projection as the scalar value.

Further, the Zernike decomposition is applied to $\rho(\mathbf{p})$, yielding a sequence of coefficients (Zernike moments) $\{\Omega_{nl}^m, l \leq n, n-l \equiv 0 \pmod 2$, where $m=0$ in 2D and $m=-l,\cdots,l$ in 3D $\}$. The sum of the squares (energy) in each band indexed by a pair of integers (n,l) is rotation invariant. Thus, $Z_{nl} = \sqrt{\sum_{i=-l}^{l}(\Omega_{nl}^i)^2}$ serves as a rotational invariant descriptor to the symmetry vector field in a local neighborhood. Because high-frequency components of the Zernike moments often contain noise, we only use the leading low frequency bands ($n \leq 8$ in 2D and $n \leq 5$ in 3D). With Zernike decomposition, the input dataset is converted into a multivariate volume, each voxel of which has a sequence of scalar values. For more details concerning 2D and 3D Zernike descriptor, please refer to [KH90] [NK03].

Although this feature descriptor is simple and independent of the mathematical model of symmetry vector field, our method identifies the features that are consistent with the analytic ones as shown in the following examples.

Visual Analysis of A 2D *N*-RoSy Field We first validate our method on 2D N-RoSy fields at the resolution of 128×128. The neighborhood size for computing the Zernike moments is set to 4. The leading 25 bands are used for analysis.

Figure 7 (a)(b)(c) show the LIC images of three different N-RoSy field fields [PZ11]. The glyphs in yellow indicate positive singularities, and the blue glyphs denote negative ones. These singularities can be the considered as ground truth for comparison. Note that the LIC images of the 3-ROSY and 6-ROSY fields are visually very similar and hard to distinguish. The results with EasyXplorer are shown in Figure 7 (d)(e)(f), in which the extracted singularities match the cases shown in Figure 7 (a)(b)(c).

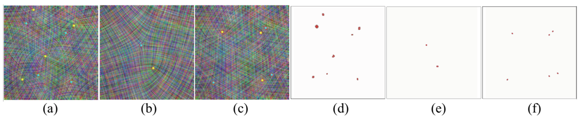

Figure 7: *Results for three 2D N-RoSy fields. (a)-(c): LIC visualization of a 3-ROSY field, a 4-ROSY field, and a 6-ROSY field, respectively; (d)-(f): The singularities discovered by using EasyXplorer.*

Visual Analysis of A Cubic Symmetry Field In our second experiment, a 3D cubic symmetry field constructed on a tetrahedral mesh is used. The field is uniformly sampled into a $100 \times 100 \times 100$ 3D grid. The neighborhood size for computing the Zernike moments is set as 4. The leading 12 bands are used for analysis. The task is to explore the patterns the cubic symmetry field may contain. Because of the feature descriptions we select, the most possible pattern is the singularity line.

At the beginning, the granularity parameter Q is set to be 16, yielding a 2D projection shown in Figure 8 (a). By slightly modulating the 2D partition, it is apparent that most data points locate in the region *reg_0_2*, and the regions corresponding to *reg_0_0* and *reg_0_1* are distributed outside of boundary in the physical space. The glyph views in *reg_0_0* and *reg_0_1* show that the statistical value of each variate in *reg_0_0* and *reg_0_1* is instable and different from those of the corresponding spatial clusters. By checking the 3D positions of *reg_0_0* and *reg_0_1* in the 3D view, the user regards them unimportant and thus removes them.

Before the next iteration, the user discovers that some data points locate near the boundary of two partitioned regions and are hard to classify. The user selects these data points (the grey circle in Figure 8 (a)). The 3D view implies that these points are also outside of the physical space. Accordingly, the user adjusts the boundary of *reg_0_2* to exclude them, regardless of *reg_0_0* and *reg_0_1*, and increases Q to 32 to take another iteration in *reg_0_2*. The result depicts that interesting patterns appear in *reg_0_0* and *reg_0_1* (Figure 8 (b)). The glyph view indicates that these two regions contain salient information that is verified by the 3D view. The user selects both regions and increases Q to 64 to further explore *reg_0_0* and *reg_0_1* (Figure 8 (c)). This exploration is iterated until a satisfying result is achieved (Figure 1).

The glyph layout widget can be used to adjust the layout of the glyph view, which offers great flexibility for highlighting and comparing characteristic attributes. As shown in Figure 1, the Z00, Z20, Z40 and Z44 in the Zernike descriptor are relatively large, while Z33 and Z53 have large variations. The visualization is consistent with the ground truth that for 3D Zernike descriptors, Z44 can be used to identify the cubic symmetry field, and Z00, Z20 and Z40 are capable of identifying the field with the radial shape. Moreover, the difference on Z33 and Z53 reveals that these regions exhibit some unusual rotation, and may indicate the singularity lines. Observing that Z33 in *reg_3_3*(pink) is larger than Z33 in *reg_3_2*(blue), and Z44 in *reg_3_3* is smaller than Z44 in *reg_3_2*, the user states that *reg_3_3* has the largest rotation. This can be confirmed by the regions in pink in the 3D view (Figure 1).

6. Evaluation

6.1. Comparisons

To some extent, our method is analogous to transfer function design schemes for volume rendering. Although we have the similar purpose and we do integrate the transfer function to highlight the clustered region, our work is very different from a transfer function designing. Take [KSC*10] as an example, the 2D/3D correlation design is quite similar to ours. However, it applies various dimension reduction approaches to highlight the ROIs by applying volume rendering, while we focus on the iterative clustering to prune the ROI gradually. [KSC*10] also adopts the interaction of selecting on

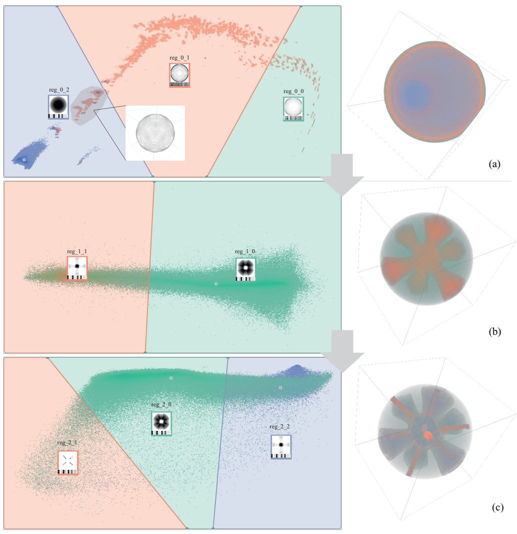

Figure 8: *The analysis process for a cubic symmetry dataset.*

2D projection to define the shown regions. We improve this scheme by integrating more visualization techniques to deliver sufficient information for decision-making.

Previous visualization systems focus on visual exploration of hidden structure, but do not incorporate the capability of heuristic data exploration. For instance, ParaView [AGL05] makes the VTK-based volume visualization scalable to large datasets, but emphasizes on the performance issues. VisIt [Vis] is an analysis tool kit for scientific visualization with professional pipeline management and parameter controlling. It is more likely a common scientific visualization framework, yet is weak in addressing the progressive exploration problem with the customized data. Our method emphasizes on the exploration with the iterative clustering mechanism, which is the main advantage over previous approaches.

6.2. Feedback from Domain Experts

We interviewed two data providers who have a deeper understanding of the data, and obtained the feedbacks to evaluate the real experience of our system. We explained to the experts our analysis pipeline, interface and analysis steps of our system, and presented the case studies. The feedbacks can be summarized as follows.

Interactive Visualization Both users agree that the method can be a useful tool for exploring multivariate spatial data. They are curious about the visual design and the interaction process. They comment that "this tool provides an interesting way to combine our knowledge with the exploring process." They like the way to interactively adjust the result and obtain the feedback in real time.

Improvements The experts comment that although the explore process is heuristic, it still needs time to try if they lack the prior knowledge. Meanwhile, it is possible that one iteration only gives locally optimal result. Thus it is necessary to keep trying and save the process of trying. Current solution only records the history of exploration by the flow chart. A better way is to design a decision tree to describe, record each branch of exploration and enable the user rollback to a step if the latest exploration attempt failed.

Besides, they are not familiar with the parallel coordinates and the glyph. Therefore it needs time for them to learn the meaning of these components. Moreover, they consider the parallel coordinates could be improved because sometimes the clusters occlude others. In fact, they would like to see the difference of two clusters on each axis. It would be better if we can highlight these differences.

7. Conclusion and Future Work

Multivariate spatial data visualization is largely motivated by the requirements of the understanding of the data distributions and investigating the inter-relationships between different data attributes. Rather than focusing on a specific technique, the presented system provides an integrated visual interface for depicting, comparing, and clustering a large amount of multivariate spatial points. As the future work is concerned, we plan to extend EasyXplorer to more types of multivariate spatial data, and parallelize the system to address even larger scale datasets. We also expect to combine well-established topological approaches into the visual exploration process for verification. Further, parameter choices and their impacts also need to be improved in the future. For example, the weight w which controls the influence of spatial position on the distribution of 2D projection. We plan to design a widget to set w as a flexible parameter on the interface. The spatial distribution encoded by a snapshot and the viewpoint is fixed along the z-axis at present. It works well in our cases, yet it may obscure information in some scenarios. Thus, a viewpoint-free snapshot will be considered in the next version.

8. Acknowledgments

This work is supported by NSFC (61232012, 61422211), Fundamental Research Funds for the Central Universities and NSF IIS-1352722.

References

[AGL05] AHRENS J., GEVECI B., LAW C.: 36 paraview: An end-user tool for large-data visualization. *The Visualization Handbook* (2005), 717. 9

[DAN*10] DANIELS J., ANDERSON E. W., NONATO L. G., SILVA C. T., ET AL.: Interactive vector field feature identification. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1560–1568. 3

[Dol07] DOLEISCH H.: SimVis: Interactive visual analysis of large and time-dependent 3D simulation data. In *Proceedings of Winter simulation* (2007), pp. 712–720. 2

[FWG09] FUCHS R., WASER J., GRÖLLER M. E.: Visual human+ machine learning. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1327–1334. 2

[GRW*00] GRESH D. L., ROGOWITZ B. E., WINSLOW R. L., SCOLLAN D. F., YUNG C. K.: WEAVE: a system for visually linking 3-D and statistical visualizations, applied to cardiac simulation and measurement data. In *IEEE Visualization* (2000), pp. 489–492. 2

[GXY12] GUO H., XIAO H., YUAN X.: Scalable multivariate volume visualization and analysis based on dimension projection and parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 18*, 9 (2012), 1397–1410. 2

[HPB*10] HAIDACHER M., PATEL D., BRUCKNER S., KANITSAR A., GROLLER M.: Volume visualization based on statistical transfer-function spaces. In *IEEE Pacific Visualization Symposium* (2010), pp. 17–24. 3

[HPC*13] HUANG J., PAN Z., CHEN G., CHEN W., BAO H.: Image-space texture-based output-coherent surface flow visualization. *IEEE Transactions on Visualization and Computer Graphics 19*, 9 (2013), 1476–1487. 3

[HTWB11] HUANG J., TONG Y., WEI H., BAO H.: Boundary aligned smooth 3D cross-frame field. In *Proceedings of SIGGRAPH Asia* (2011), pp. 143:1–143:8. 3, 7

[JBS08] JANICKE H., BOTTINGER M., SCHEUERMANN G.: Brushing of attribute clouds for the visualization of multivariate data. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1459–1466. 2

[JCC*11] JOIA P., COIMBRA D., CUMINATO J. A., PAULOVICH F. V., NONATO L. G.: Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2563–2571. 4

[KASH13] KRATZ A., AUER C., STOMMEL M., HOTZ I.: Visualization and analysis of second-order tensors: Moving beyond the symmetric positive-definite case. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 49–74. 2

[KERC09] KEEFE D., EWERT M., RIBARSKY W., CHANG R.: Interactive coordinated multiple-view visualization of biomechanical motion data. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1383–1390. 2

[KH90] KHOTANZAD A., HONG Y. H.: Invariant image recognition by zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 5 (1990), 489–497. 7, 8

[KH13] KEHRER J., HAUSER H.: Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Transactions on Visualization and Computer Graphics 19*, 3 (2013), 495–513. 2

[KSC*10] KIM H. S., SCHULZE J. P., CONE A. C., SOSINSKY G. E., MARTONE M. E.: Dimensionality reduction on multi-dimensional transfer functions for multi-channel volume data sets. *Information visualization 9*, 3 (2010), 167–180. 2, 3, 8

[LHD*04] LARAMEE R. S., HAUSER H., DOLEISCH H., VROLIJK B., POST F. H., WEISKOPF D.: The state of the art in flow visualization: Dense and texture-based techniques. In *Computer Graphics Forum* (2004), vol. 23, pp. 203–221. 2

[LYL*06] LUNDSTRÖM C., YNNERMAN A., LJUNG P., PERSSON A., KNUTSSON H.: The alpha-histogram: Using spatial coherence to enhance histograms and transfer function design. 227–234. 3

[MJW*13] MACIEJEWSKI R., JANG Y., WOO I., JANICKE H., GAITHER K. P., EBERT D. S.: Abstracting attribute space for transfer function exploration and design. *IEEE Transactions on Visualization and Computer Graphics 19*, 1 (2013), 94–107. 3

[MM09] MUELDER C., MA K.-L.: Interactive feature extraction and tracking by utilizing region coherency. In *PacificVis* (2009), IEEE, pp. 17–24. 3

[NK03] NOVOTNI M., KLEIN R.: 3d zernike descriptors for content based shape retrieval. In *ACM Symposium on Solid modeling and applications* (2003), pp. 216–225. 7, 8

[NN04] NOCK R., NIELSEN F.: Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 11 (Nov. 2004), 1452–1458. 3, 4, 6

[PVH*03] POST F. H., VROLIJK B., HAUSER H., LARAMEE R. S., DOLEISCH H.: The state of the art in flow visualisation: Feature extraction and tracking. In *Computer Graphics Forum* (2003), vol. 22, pp. 775–792. 2

[PZ07] PALACIOS J., ZHANG E.: Rotational symmetry field design on surfaces. In *ACM Transactions on Graphics (TOG)* (2007), vol. 26, ACM, p. 55. 7

[PZ11] PALACIOS J., ZHANG E.: Interactive visualization of rotational symmetry fields on surfaces. *IEEE Transactions on Visualization and Computer Graphics 17*, 7 (2011), 947–955. 3, 8

[RBS05] ROETTGER S., BAUER M., STAMMINGER M.: Spatialized transfer functions. In *EuroVis* (2005), pp. 271–278. 3

[RP08] ROPINSKI T., PREIM B.: Taxonomy and usage guidelines for glyph-based medical visualization. In *Proceedings of Simulation and Visualization* (2008), pp. 121–138. 2

[TPM05] TORY M., POTTS S., MÖLLER T.: A parallel coordinates style interface for exploratory volume visualization. *IEEE Transactions on Visualization and Computer Graphics 11*, 1 (2005), 71–80. 2

[UIM*03] URNESS T., INTERRANTE V., MARUSIC I., LONGMIRE E., GANAPATHISUBRAMANI B.: Effectively visualizing multi-valued flow data using color and texture. In *IEEE Visualization* (2003), pp. 115–121. 2

[Vis] VISIT:. http://wci.llnl.gov/simulation/computer-codes/visit. 9

[WZK12] WANG L., ZHAO X., KAUFMAN A. E.: Modified dendrogram of attribute space for multidimensional transfer function design. *IEEE Transaction on Visualization and Computer Graphics 18*, 1 (2012), 121–131. 2, 3

[ZBDS12] ZINSMAIER M., BRANDES U., DEUSSEN O., STROBELT H.: Interactive level-of-detail rendering of large graphs. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2486–2495. 2

[ZK10] ZHAO X., KAUFMAN A.: Multi-dimensional reduction and transfer function design using parallel coordinates. In *IEEE/EG international conference on Volume Graphics* (2010), pp. 69–76. 2

[ZTM*13] ZHANG Z., TONG X., MCDONNELL K. T., ZELENYUK A., IMRE D., MUELLER K.: An interactive visual analytics framework for multi-field data in a geo-spatial context. *Tsinghua Science and Technology 18*, 2 (2013), 111–124. 2