Visual Exploration of Spatio-temporal Relationships for Scientific Data

Sameep Mehta*
Computer Science & Engineering
The Ohio State University

Srinivasan Parthasarathy
Computer Science & Engineering
The Ohio State University

Raghu Machiraju
Computer Science & Engineering
The Ohio State University

ABSTRACT

Spatio-temporal relationships among features extracted from temporally-varying scientific datasets can provide useful information about the evolution of an individual feature and its interactions with other features. However, extracting such useful relationships without user guidance is cumbersome and often an error prone process. In this paper, we present a visual analysis system that interactively discovers such relationships from the trajectories of derived features. We describe analysis algorithms to derive various spatial and spatio-temporal relationships. A visual interface is presented using which the user can interactively select spatial and temporal extents to guide the knowledge discovery process. We show the usefulness of our proposed algorithms on datasets originating from computational fluid dynamics. We also demonstrate how the derived relationships can help in explaining the occurrence of critical events like merging and bifurcation of the vortices.

Keywords: Knowledge Discovery, Scientific Analytics, Trajectory Analysis, Feature Extraction, Spatio-temporal Predicates, Visual Analytics

Index Terms: D.2.2 [Design Tools and Techniques]: User Interfaces; H.2.8 [Database Applications]: Data Mining; H.2.8 [Database Applications]: Scientific Databases;

1 INTRODUCTION

The physical and engineering sciences are interested in the study of large, time varying scientific datasets to capture and understand the underlying physical process exhibited by intrinsic features present in the datasets. The features often evolve and interact with other features resulting in important phenomena and critical events. For example, interactions among the anomalous structures in molecular dynamics simulation result in long extended defects structures [18], which affect the electrical and mechanical properties of the semiconductor in an undesirable fashion. To mitigate this effect, the doping process and the associated parameters should be chosen carefully. Similarly, studying the interactions among the vortices in computational fluid dynamics help in better design of airplane wings. In this paper, we employ various spatial and spatio-temporal relationships to capture and understand the evolution of an individual feature and the interactions among several features. The information provided by various spatio-temporal relationships can be used to understand the effect of initial simulation parameters on the behavior of the features and can help select the parameters carefully to steer the simulations. Additionally, the simulation parameters can be changed in a controlled fashion. By observing the corresponding changes in different relationships, the effect of the parameters on the behavior of the features can be studied in a systematic fashion. These relationships also help in overall knowledge mining process from such datasets. For example, the (most likely) process resulting in critical events like merging, bifurcation etc can be easily understood. Furthermore, important components of the data which require more detailed analysis can be easily identified, resulting in huge reduction in time and computational power. The overall goal of efficiently deriving meaningful relationships among scientific features is hindered by two challenges i) finding an useful representation for the evolving features and ii) effectively selecting the spatial region and temporal interval to be used in establishing the relationships. Next, we discuss these challenges in detail and also highlight the proposed solutions to address these issues.

A fundamental property of spatio-temporal features is that: *the spatial positions of the features change over time*. This change in position can be characterized by motion parameters including linear velocity and angular velocity. Moreover, in scientific datasets, the extent and the shape of the features also change frequently. This information is very important to precisely describe the evolution of the feature and should be taken into account in the analysis. We capture the change in the size of the feature by using scaling parameters. The change in these spatial properties can lead to interesting phenomena like dissipation and creation of new features. For example, a shrinking feature may cease to exist at some later time instant. The evolution of the individual features can also induce changes in the relationships among features e.g. two far apart features moving toward each other will most likely interact at some future time instant. These interactions result in the occurrence of the critical events including merging and bifurcation [25]. We use the following attributes to represent the trajectory of a scientific feature: *i) position, ii) change in position over time, iii) extent and iv) change in extent over time.* Interested readers are referred to our previous work [19] for a complete description of the representation scheme. Our analysis component makes use of this representation to precisely identify the spatial and spatio-temporal relationships.

Another important characteristic of the objects [1] is that most of the time they interact exclusively with other objects present in the neighboring spatial region(s). Correct identification of such regions is an extremely important step toward deriving meaningful spatio-temporal relationships. There are infinite possibilities for selecting the size and position of such region(s). Choosing a large region will result in deriving non-existent relationships. Similarly, choosing a small region will result in missing important interactions. The size and position of these regions differ not only across datasets but also varies across different features present in the same dataset. The problem becomes even more intricate when the features are moving and changing extents. In such cases, one single method to define the neighborhood for all the features can produce sub-optimal or even wrong results. Similarly, determining a useful time interval for the analysis poses yet another challenge to the knowledge mining process. Incorrect intervals can potentially lead to incomplete results, e.g., a very interesting phenomenon will be missed if it occurs just after or before the selected time interval. Therefore, we contend that each feature in the dataset requires individual attention while selecting meaningful regions and time intervals to be used for further analysis. Therefore, there is strong need for a visual interface though which the user can interactively select the extents based on the characteristics of other features and the final goal. The trajectories of different features often overlap therefore, the user should

*{mehtas,srini,raghu}@cse.ohio-state.edu

[1] In this article we use feature and object interchangeably

IEEE Symposium on Visual Analytics Science and Technology 2006
October 31 - November 2, Baltimore, MD, USA
1-4244-0592-0/06/$20.00

also possess the capability to focus on a smaller part of the trajectory while hiding the other parts. Finally, as discussed earlier, the extent of the object plays a very important role in the analysis process. However, displaying the extents of all the objects for the whole time span of the simulation will result in a highly cluttered visual representation. It will be extremely difficult to glean any useful information from such a cluttered view. Therefore, the extents are displayed only on the user's request. These features together with the interactive selections form the visualization component of the proposed system.

To summarize, we describe a visual reasoning and knowledge mining system to understand the spatial and spatio-temporal relationships among evolving features. In this article, we focus on vortices extracted from CFD data. Both the, analysis and visualization, components help the users to discover useful information from the datasets efficiently by making the search process and reasoning more focused and goal driven. The visualization component enables the user to interactively select meaningful spatial and temporal extents to perform the analysis on. The analysis results provide useful information about the behavior of features w.r.t. the selected extents. This process is repeated till the user discovers the information he or she is seeking or finds new information.

To summarize, the key contribution of this article are:

1. We present an interactive visual interface allowing the users to select spatial and temporal extents. The interface also supports the *zoom, filter and details on demand* paradigm [3].

2. We present algorithms for automatically deriving various spatial and spatio-temporal relationships including the topological relationships proposed by Egenhofer [5].

3. We empirically demonstrate the usefulness of our algorithms on datasets originating from CFD. We also discuss the use of domain knowledge coupled with our system to derive useful information from these trajectories.

The rest of the article is structured as follows: In Section 2 we review some of the existing research that is related to this work. Section 3 presents the important components of the proposed system. Section 4 describes our motion representation, analysis and visualization algorithms in details. Results on simulation datasets are shown in Section 5. Finally, we discuss some of our ongoing and planned initiatives for this problem in Section 6

2 Related Work

Our motion estimation algorithm is closely related to the trajectory representation schemes present in the existing literature of data mining and databases. These schemes can be divided into two broad categories i) native (*xyz*) space representation and ii) parametric space representation. For an excellent survey on most popular native space representation techniques the readers are referred to Mokbel *et al.* [20]. Techniques which exploit one dimensional time series including those based on DFT [1, 9], DWT [21], SVD [16] are not directly applicable in this context. Kollios *et al.* [15], Saltenis *et al.* [24] and Tao *et al.* [26], represented the trajectories by sub-segments such that the object moved with same velocity in a sub-segment. Storing the linear velocities instead of object locations result in efficient query processing and low overheads in maintaining and updating the index structure. All these approaches abstract the object by a point (typically center of mass). This simplification produce highly efficient algorithms. However, it misses crucial information. A extent- and shape-aware based distance calculation between 2 objects is much more meaningful than the one based on just the center of mass. Moreover, given the center of mass of an object at two successive time instants, the translation matrix (and hence the linear velocity) which optimally maps one point to another can be derived. Estimation of both angular and linear velocity from two points is an ill-posed problem. Additionally, since only points are considered, object scaling is not defined. Therefore, by using point based representation we cannot completely characterize the motion.

Egenhofer [5], presented a 9-intersection model to establish topological relationships like *meet, inside, overlap* between 2D objects. The model finds the relationships by considering 9 possibilities between boundary, interior and exterior of one object with the corresponding parts of the other object. Erwig and Schneider [8], extended these ideas to describe spatio-temporal predicates. The authors described 8 basic spatio-temporal predicates like *disjoint, inside, meets*. Recently, Erwig and Schneider also presented some guidelines on the representation of a sequence of spatio-temporal predicates [7]. The authors assumed that the spatial and temporal extents which can result in useful relationships are available. The focus of their work was on defining the relationships not on choosing the extents.

Recently, we [27, 28], presented algorithms for mining frequent spatial patterns from scientific datasets. The main goal of that work was to find spatial patterns and use that information to reason about the critical events. Study of the motion of individual objects was not performed. Additionally, *navigational*, *topological*, *directional* and *interaction* analysis was not discussed in this previous work.

Hochheiser and Shneiderman [11], presented a tool TIMESEARCHER for visualizing and interactively querying time series datasets. Recently Lin *et al.* [17], proposed VizTree for pattern discovery, anomaly detection and querying in large scale time series datasets. Chittaro and Combi [4], presented different approaches for representing temporal relations. However, these tools were developed for primarily for analyzing time series data. Hamarneh and Gustavsson [10], presented algorithms for modeling and segmenting 2d time varying shapes. However, no explicit modeling of motion parameters was performed. Eickhorst *et al.* [6], proposed spatio-temporal helix to model the trajectory of the object. The authors demonstrated the use of this representation for comparing two trajectories. However, visualization and analyzing relationships among objects was not the focus of that work [10, 6].

3 Overview and Background

Figure 1 schematically describes our proposed system. The main components of the system are:

- **Analysis1 - Data Transformation:** This component primarily deals with transforming the simulation data into a format which can be used for visualization and knowledge discovery. The process starts by extracting meaningful features (regions of interest) from scientific datasets. The trajectory of each temporally varying feature is represented by a set of non-overlapping temporal segments. Within each segment important motion parameters including linear velocity $\vec{v}$, angular velocity $\vec{\omega}$ and scaling parameters $\vec{s}$ are estimated [19]. More details on parameter estimation and segmentation algorithms can be found in [19]. Apart from being physically meaningful, this representation also reduces memory overheads. Moreover, important characteristics about the motion of a feature can also be ascertained by analyzing the motion parameters.

- **Visualization - User Interface:** The segmented trajectories are then visualized for further analysis. The user can interactively define spatial and temporal extents. The selected extents are used for establishing various spatial and spatio-temporal relationships. The user can *zoom* and *filter* the trajectories to focus on the most interesting and important parts. Finally, more details about the objects can be accessed, if needed.

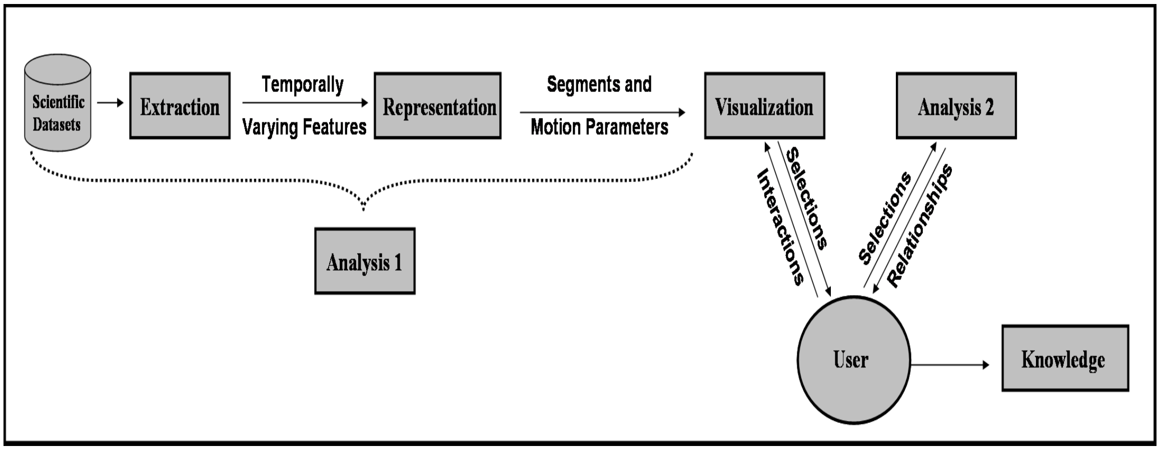

Figure 1: Overview of the System for Understanding Trajectories of Scientific Objects

- **Analysis2 - Deriving Relationships:** This component is the backend engine for establishing various relationships. In this paper, we focus on directional, navigational and topological relationships. Based on the high level relationships the user can refine spatial and temporal extents to obtain more detailed information. The algorithms can also help explain the likely cause of critical events like bifurcation and merging.

The user iterates through components 2 and 3, reducing the search space in each iteration to obtain more detailed information about the feature. Our previous research efforts have largely focused on the first component of the system [12, 13, 18]. Therefore, in this work, we concentrate on the other components. However, to make this paper self-contained, we briefly describe our previous work as it is related to this work. Jiang *et al.* [12] presented a general framework for feature extraction from scientific datasets. We showed the usefulness of the framework on datasets originating from computational fluid dynamics and computational molecular dynamics. In this paper we use the algorithms presented by Jiang *et al.* [12, 13] for extracting vortices from temporally varying fluid flow datasets. Recently, we proposed a parametric scheme for representing the motion of evolving features [19]. Our representation scheme is based on estimating important motion parameters including linear velocity and angular velocity. The change in the size of the object is characterized by scale parameters. All these parameters together are referred to as Motion Parameter Vector (MPV). Least square error was minimized to estimate the parameters between every two consecutive frames in the dataset. Next, the trajectories were segmented into piecewise smooth sub-trajectories using a clustering algorithm. The clustering algorithm uses the estimated MPV as a feature vector. We use weighted Euclidean distance as the distance metric in the clustering algorithm. Each sub-trajectory is represented by a single MPV. We strongly believe that this representation is physically meaningful. The representation also results in high compression ratio which makes it useful for large scale simulation datasets. Additionally, the representation also lends itself for prediction algorithms. We showed the effectiveness of this representation for prediction and analysis for datasets originating from various domains. Please note that in the previous work [19], *no visual component* was presented. The focus there was to motivate the need and evaluation of this representation. Most of the analysis reported in [19] was performed by trying several spatial and temporal extents. *The cumbersome manual process motivated us to develop this visual interface.*

4 ALGORITHMS

In this section we present the last two components of the above mentioned system in detail. First, the basic notation used in the paper is presented.

Basic Notation: S denotes a time varying dataset with N steps monitoring the movement of m objects $O = \{O_1, O_2, \ldots, O_m\}$. An object O_r is represented by K points (landmarks) sampled from the surface of O [27, 22]. At the i^{th} time step the state of O_r is represented by $O_{r,i} = [\{x^1_{r,i}, y^1_{r,i}, z^1_{r,i}\} \ldots, \{x^k_{r,i}, y^k_{r,i}, z^k_{r,i}\}]$. The position of the j^{th} landmark at the i^{th} time step is denoted by $O^j_{r,i}$. The time between two successive time steps is denoted by δ. After determining motion parameter vectors (MPV) and segmentation, the j^{th} sub-trajectory of O_r is denoted by $O_{r,j}$ and is represented by the following feature vector

$$\{[t^j_{r,1}, t^j_{r,2}], [\{x^1_{r,t^j_1}, y^1_{r,t^j_1}, z^1_{r,t^j_1}\} \ldots, \{x^k_{r,t^j_1}, y^k_{r,t^j_1}, z^k_{r,t^j_1}\}], \{P^1_{r,j}, P^2_{r,j} \ldots P^M_{r,j}\}\}$$

where $\{P^1_{r,j}, P^2_{r,j} \ldots P^M_{r,j}\}$ represents the MPV of the j^{th} sub-trajectory of O_r. The time interval of the j^{th} segment is $[t^j_{r,1}, t^j_{r,2}]$. The stored $\{x, y, z\}$ points are landmarks describing the shape of the object at start of the segment i.e at time $t^j_{r,1}$. Description of the feature anywhere else in j^{th} segment is obtained by simply applying the motion parameters $\vec{v}$, $\vec{\omega}$ and $\vec{s}$ to the shape descriptor.

Next, we describe the main analysis tasks and relationships. For expository simplicity, we present the analysis component first so that the motivation behind the design of visual component and user interactions can be explained clearly.

4.1 Analysis2 - Deriving Relationships

The set of m objects is denoted by $O = \{O_1, O_2, \ldots, O_m\}$. The time interval is denoted as $t = [\, t_s, t_e]$ where $t_s \leq t_e$. A single time instant is denoted by t_l. Furthermore, R represents a spatial region with $[R_{lx}, R_{ly}, R_{ux}, R_{uy}]$ denoting the lower (l) and upper (u) co-ordinates (x, y) of R. In this section we focus on the deriving the following spatial and spatio-temporal relationships.

- **Directional Relationships -** These relationships provide information about the spatial location of an object with respect to the other objects in O. A typical query in this scenario is *where is object $O_{r,l}$ located wrt to $O_{s,l}$ at time t_l?* We use four operators *left, right, top and bottom* to characterize this relationship. The actual relations are established by comparing the K landmark points (explained in last section). These simple operators are then combined to derive advanced relationships like *top-left* and *bottom-right* etc.

- **Topological Relationships -** Topological relationships help identify the connection between a region R and the object O_r at a time instant t_l (denoted by $O_{r,l}$). We characterize these relationships by *inside, outside and overlaps* operators. The object $O_{r,l}$ is said to be *inside* (*outside*) R if the whole object is enclosed (not included) in R. The *overlap* is defined if some part of $O_{r,l}$ is inside R. In our representation, we check the K landmark points of $O_{r,l}$ against R. Let $IO(p,R)$ be an **I**nside-**O**utside test which returns $true$ if the point p lies inside R, $false$ otherwise. This function can be trivially implemented by checking the x and y value of p against $[R_{lx}, R_{ly}, R_{ux}, R_{uy}]$. Given this function, various topological relationships can be derived as follows:

$$O_{r,l} \text{ is } \textit{inside } R \text{ if } \forall i \in [1,K]\ IO(O^i_{r,l}, R) = true$$
$$O_{r,l} \text{ is } \textit{outside } R \text{ if } \forall i \in [1,K]\ IO(O^i_{r,l}, R) = false$$
$$O_{r,l} \textit{ overlaps } R \text{ if } \exists i \in [1,K]\ IO(O^i_{r,l}, R) = true$$

The above described topological relationships are defined only for a single time instant. In case of an evolving feature, we characterize the spatio-temporal topological relationships by five events *enter, leave, disjoint, cross and contain*. O_r is said to have *entered* (*left*) R between $[t_s, t_e]$ if $O_{r,s}$ was outside and $O_{r,e}$ was inside, i.e., we check the location of O_r at the start and the end of the time interval. Similarly, if both $O_{r,s}$ and $O_{r,e}$ are inside R, then the *contain* relationship is identified. However, to distinguish *disjoint* and *cross* events we need to process every time step because $O_{r,s}$ and $O_{r,e}$ will be outside R, in both the cases. These relations are derived by using the above described *inside, outside and overlap* operators:

O_r *entered* R in $[t_s, t_e]$ if $O_{r,s}$ is *outside* R && $O_{r,e}$ is *inside* R

O_r *left* R in $[t_s, t_e]$ if $O_{r,s}$ is *inside* R && $O_{r,e}$ is *outside* R

O_r *contain* in R in $[t_s, t_e]$ if $O_{r,s}$ is *inside* R && $O_{r,e}$ is *inside* R

O_r *crossed* R if $O_{r,s}$ and $O_{r,e}$ are outside R && $\exists i \in [t_s, t_e]$ $O_{r,i}$ is *inside* R

O_r is *disjoint* with R if $O_{r,s}$ and $O_{r,e}$ outside R && $\forall i \in [t_s, t_e]$ $O_{r,i}$ *outside* R

- **Navigational Relationships -** This analysis is used to understand the motion characteristics of the objects. First the user selects a spatial region and time interval and all the motion parameters of the trajectories are displayed. Important characteristics of the features can be easily ascertained by inspecting the associated MPVs. For example, positive trend in speed implies accelerating object. Similarly, the angular velocity helps to check if the object is rotating in the clockwise or counter clockwise direction. Similarly, scaling parameters is less that 1 implies a shrinking feature.

  Next, we use the *follow* operator described by Roddick *et al.* [23]. Given a spatial region R (and a time interval), the operator established if within R a feature demonstrates similar motion to itself or to other features. Using our representation, we first find the motion parameters for each connected trajectory. Next, the distance between these parameters is calculated. If the distance is less than a $user\ defined\ threshold$, then the trajectories are said to *follow* each other. *Please note establishing this relationship is very similar to the problem of finding matching sub-trajectory in database community [9].*

The results of the analysis can also be viewed in form of a animation, which provides more details about the behavior of the object at each time step. We discuss this aspect in detail in Section 5.

4.2 Visualization - User Interface

Although the analysis component appears to be self-sufficient to understand the evolutionary behavior of the object, however the main problem is how to select potentially useful spatial and temporal extents? Without any visual aids, this problem requires a brute force algorithm. For example, assume that *we want to find the largest region R such that no object entered R in* $[t_s, t_e]$. Such an R can provide valuable information about the underlying physical parameters which makes R unconducive for any object's movement through it. The region R can be found by performing an exhaustive search over the whole space changing the size, orientation and position of R in every iteration. This process is computationally prohibitive. However, with a visual interface the user can start by defining a coarse region first and refine it by changing the size and orientation to find an appropriate R. Therefore, the user can identify the potential regions very quickly making the search process more focused, efficient, and meaningful.

In this section we describe the visual representation we use. Specifically, we employ two graphs i) spatial graphs (SG) and ii) temporal graphs (TG) for representing spatial and temporal information of the trajectories respectively. Next, we explain each of the graphs and associated user interactions in detail and also point to the use of these graphs for visual analysis and reasoning.

- **Spatial Graphs (SG):** This graph displays the trajectories in xy space. Different colors are used for different trajectories. For clarity, SG only shows the point trajectory of the objects. These point trajectories are computed by recording the position of center of mass of the object at each time step. The user can access more details by requesting the system to display the extents and shape of the objects.

- **Temporal Graphs (TG):** This graph describes the temporal behavior of the objects. For each object the life time (the time for which the object existed) is divided into subsegments. The length of the subsegments is again specified by the temporal range of each sub-trajectory. The sub-trajectories are obtained by MPV estimation and clustering algorithm [19].

- **User Interactions:** For the analysis tasks we need a spatial region R and a time interval $[t_s, t_e]$. The user interactively selects the spatial region R and temporal extents $[t_s, t_e]$. The parts of the trajectory that lie inside R and are active during $[t_s, t_e]$ are highlighted in real time. The user can then choose to zoom all the sub trajectories within R. The user also has the capability to hide some of the trajectories and focus on the visually more interesting trajectories. Once the user is satisfied with spatio-temporal extents, he or she can start the analysis by invoking function calls to the backend engine. The results of the analysis are visually presented to the user. The results can be displayed either statically or as a animation. Based on the results, the user refines the search space and again engages the analysis tools. This iterative process is continued until the final desired information is extracted. The user can not only iterate between the visualization and analysis components, but also can switch among various analysis components. For example, combining results from topological relations and navigational relations can help to predict if the objects will start interacting in the near future. This can be done by finding two spatially proximate objects which are moving toward each other. Spatial proximity is ascertained by topological relationships and the direction of the movement is found using the navigational relationships.

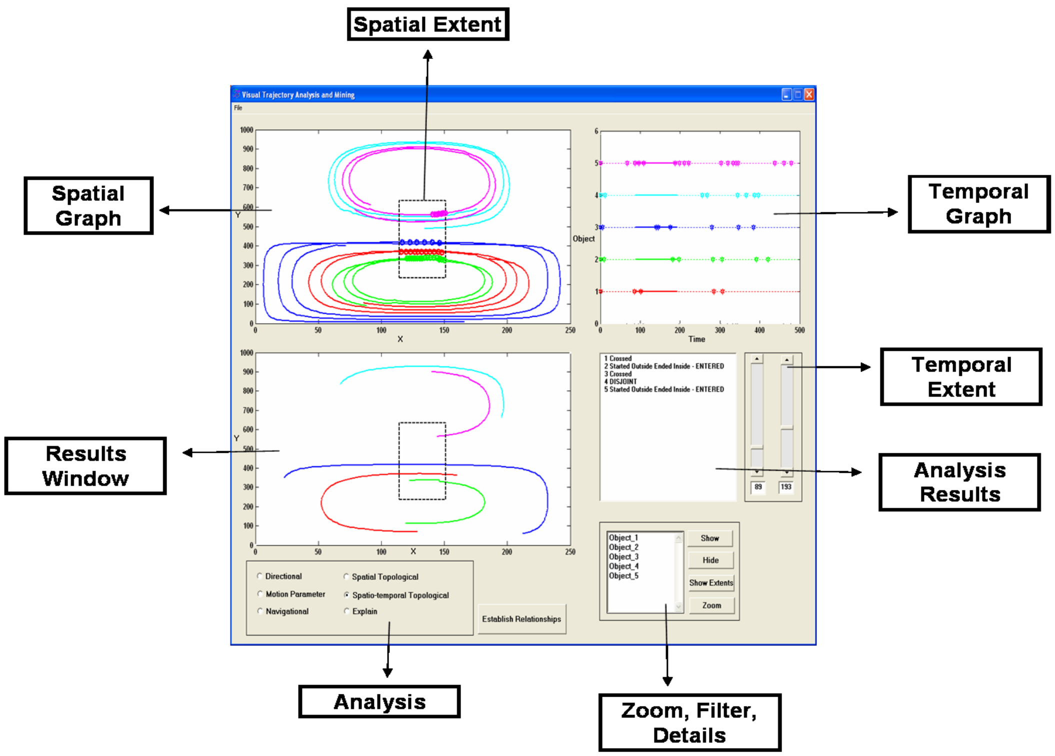

Figure 2: Overview of the Visual Interface

Now, we present an overview of our visual interface, highlighting the use of the major parts of the interface. Figure 2 show one snapshot of the visual system. The top two graphs are Spatial Graph (SG) and Temporal Graph (TG) respectively. The same color is used for the objects in all the graphs to establish correspondence. The markers on TG indicate the segment boundaries. *Please note that the length of some intervals seems to be* 1*, however this is not the case. These intervals are small and represent large change in the motion.* The black rectangle shows the user specified Spatial Region (R). The sliders shown in Figure 2 are used to select the Temporal Extent ($[t_s, t_e]$. If the relationships are defined only for a single time instant either both the sliders are set to the same value or the second slider is simply ignored. The *Zoom* operation is handled by the lower right frame. This frame also supports *Filter* operations. The user can select object(s) and choose to hide (show) them. Similarly, more *Details* are accessed by displaying the extents of the objects. The lower right frame shows all the operations which our system currently supports. The Result Window (RW) visually displays the results of the analysis. These results, along with more information, are displayed in plain text format in the Analysis Result (AR) window.

5 Results

In this section we demonstrate the use of our system on datasets originating from computational fluid dynamics. We used the simulation model proposed by Kim and Machiraju [14] to generate the datasets. The features (vortices) are detected by using the algorithms proposed by Jiang *et al.* [13]. Each vortex is approximated by an ellipse. Next, 10 points (landmarks) are sampled from the boundary of the ellipse. Finally, MPVs are estimated and the trajectory is segmented. This representation is the input for visualization and analysis components.

Spatial Topological Relationships- Figure 3(b) shows the derived topological relationships. Since this class of relationships is defined only for a single time instance, we only consider the first slider. RW shows the selected region R and the position of the features at $t_l = 154$. AR displays the computed relationships. Objects 1 and 4 are determined to be *outside* R whereas objects 2 and 3 are *inside* R. Object 5 *overlaps* R.

Spatio-Temporal Topological Relationships- Figure 2 is an example of spatio-temporal topological analysis. The RW (Result Window) displays the parts of trajectories which are active during the selected time interval. In the SG the parts of trajectory which are active during time interval and lie in the selected spatial region R are highlighted. Similarly, the time interval is highlighted in TG (Temporal Graph). The derived relationships are shown in AR (Analysis Result) window. For example, objects 1 and 3 *crossed* R. Similarly, object 4 is *disjoint* with R and objects 2 and 5 *entered* R.

Directional Relationships- Figure 3(a) shows the derived directional relationships. In this case, we decided to concentrate only on objects 3, 4 and 5. Other two objects are hidden. Since this class of relationships is defined only for a single time instance, only first slider is considered. Additionally, R is not needed for directional relationships. Please note absence of R is consistent with our definition of directional analysis. R can be easily accommodated by considering only the features which are inside R. Such features can be identified using spatial topological relationships. RW shows the position and orientation of vortices at selected time instant ($t_l = 179$). AR displays the computed relationships. For example, object 3 (blue color) is to the LEFT and BOTTOM of other two objects (4 and 5). Please note that if object 1 is to the left of object 2 then, object 2 is to the right of object 1. Due to this property, we report only one relation between two objects. The other relation is trivially derived.

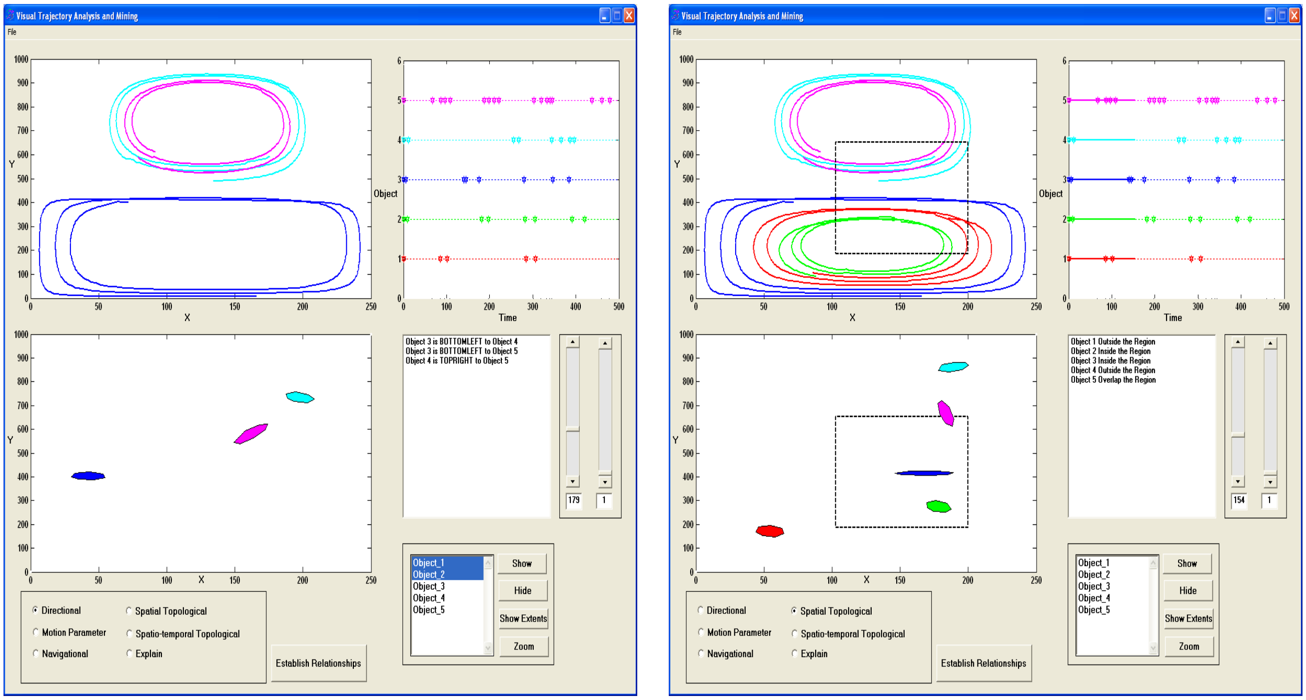

Figure 3: (a) Directional Relationships (b) Spatial Topological Relationships

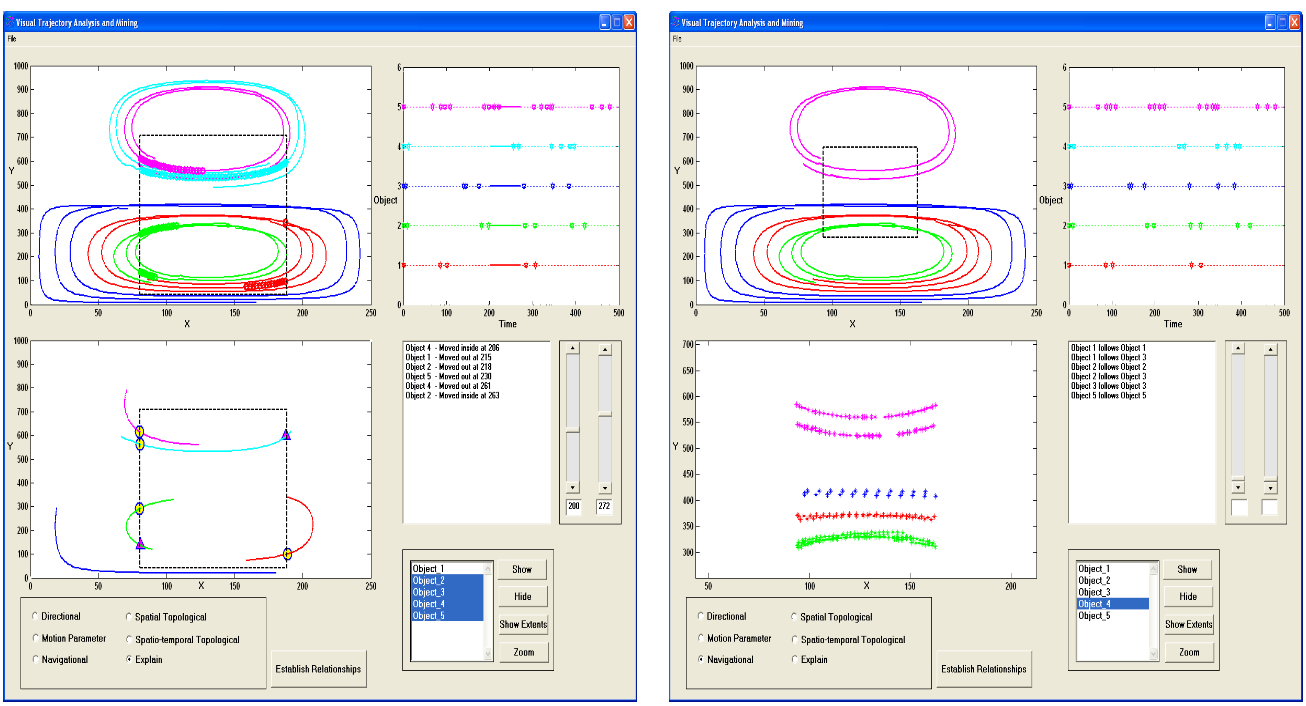

Figure 4: (a) Explain Mode (b) Navigational Analysis

Explain Mode- Figure 4(a) shows an example of the *explain mode*. This functionality is added to extract detailed information, if needed, from the analysis. *RW* shows the selected trajectories. Different markers are used to highlight the entrance (exit) of features in *R*. The information along with time instances is shown in *AR*. Spatio-temporal analysis provides the information by deriving relations like *enter, inside* etc. These relationships are established by just checking the location of the object at the start and the end of the time interval. Although the relationships are derived very efficiently, they can sometimes provide incomplete results, e.g. in figure 4(a) object 2 started *inside R* and ended *inside R*, topological analysis will return *contain* as the relationship. However, by using the explain feature we can easily determine that the object moved *outside* at $t = 218$ and moved back in at $t = 263$. This mode is intended to be used in conjunction with topological analysis to provide more detailed answers. The user first performs topological analysis, hides the uninteresting objects (objects 4 and 5 in this example) and refines the extents. When the user is satisfied with the extents, more information about the interesting objects is obtained by using this mode. The information can then be used to construct temporal rules. For example, the rules generated for objects 1 and 2 are:

Object 1 is *Inside* between $[200,215]$
Object 1 is *Outside* between $[216,262]$
Object 2 is *Inside* between $[200,218]$
Object 2 is *Outside* between $[219,262]$
Object 2 is *Inside* between $[263,272]$

These rules can be used to mine temporal relations among different objects by using Allen's temporal algebra [2]. Allen [2] describes 13 relationships including *before, after, contained by* etc which can exist among temporal intervals. An example of such a rule will be *Object* 1 *inside during Object* 2 *inside*, implying that whenever object 1 was *inside* object 2 was also *inside*. We are currently investigating algorithms for efficient mining of all such rules. Additionally, the explain mode redraws the plot at every time step making it relatively slower. Therefore, using it instead of topological analysis is not recommended.

Navigational Analysis- Figure 4(b) demonstrates the use of navigational analysis. We only need to define a spatial region for these relationships. *RW* shows the zoomed view of the trajectories in the selected spatial region. *AR* displays the final relationships. Please note that object 4 is hidden. Object 5 is found to *follow* itself i.e. the object shows a similar motion which it displayed in some other time interval. Similarly, object 1 is following itself and object 3. Please note that *follows* is a symmetric relation between two objects. Therefore, the relationship is displayed only once.

Discovering Interesting Spatial Regions- Our system can be used to interactively discover interesting regions in the dataset. We present one such example in Figure 5(a). The goal here was to find the largest spatial region *R* such that no vortex was present in *R* given a time interval. If the interval covers the entire span of the simulation, then presence of such an area suggests that the initial simulation parameters does not allow any vortex to enter this region. Figure 5(a) shows such an area and also the extents of the objects. The initial selection was made by observing the empty space in *SG*. We ran our analysis algorithms on the selected region. In first few attempts, we found that even though no object entered *R* (spatio-temporal topological analysis), but some objects were overlapping (through spatial topological operations). Based on these results we successively refined the area, until no object entered or overlapped with *R*. We were able to find *R* after 5 iterations of refinement.

Explaining Critical Events- Figure 5(b) show the output of our system on another dataset. First of all, from the TG we can learn that object 2 ceases to exist at $t = 70$ and objects 5 and 6 are created at $t = 75$. Most likely, object 2 bifurcates into objects 5 and 6. Next, we tried to understand the the process which can explain this event. We selected an *R* around object 2 and the time interval is selected as $[55,80]$. By using the spatio-temporal analysis and explain mode, we found that object 1 entered *R* at t= 58 and started interacting with object 2. At $t = 64$, the distance between objects 1 and 2 was very small, indicating stronger interactions. Finally, at $t = 70$, object 2 splits into object 5 and 6. The whole process of bifurcation takes place in interval $[68,75]$. During the interval $[71,74]$, the shape of the object 2 was deformed in such a fashion that it cannot be precisely represented by an ellipse. Therefore, in *SG* we see a large variation in the position of the center of mass of object 2 (green color). This also explains why we don't see a single curve splitting into two curves clearly.

6 Conclusions and Discussion

In this article we presented a visual analysis system for knowledge discovery from time varying scientific datasets. Motion parameters are used to represent the trajectories of the features. The parametric trajectories are presented visually to the user. The user interacts with the visual interface and invokes the analysis engine to extract spatial and spatio-temporal relationships of interest.

Currently, we are extending the framework to incorporate a prediction module which will predict not only the positions of the objects but also the most likely interactions among the objects. We are also investigating efficient algorithms to derive temporal relations [2] and convert them to visual representation. The other aspect we would like to address is to handle streaming datasets. Most of our analysis algorithms are fast enough to obtain real time performance. Therefore, we believe that it should not be too difficult to extend our framework for applications requiring analysis of streaming data.

Although the analysis algorithms can be trivially implemented to handle 3D trajectories, it is not straightforward to extend the interactive part of visual component to do the same. The main challenge is to select a 3D region by using the 2D display and mouse in an efficient and user friendly manner. We have identified two different approaches to address this issue. The first method is via projections, i.e., the 3D data can be projected onto *xy*, *yz* and *xz* planes and the region can be selected by using two of the three views. The second method uses rotation. First, two dimensional region *R* is selected using the current approach. Next, the display is rotated by 90 degrees and a spatial extent is selected in the third dimension. Finally, a 3D region can be constructed by using the two dimensional region and the spatial extent in the third dimension. In the future, we plan to incorporate both these methods in the toolkit and perform a user study to select the most viable option.

7 Acknowledgments

This work is funded by the following NSF grants NGS-0326386, ACI- 0234273 and Career Award IIS-0347662. The authors would like to thank Professor David Thompson and Monika Jankun-Kelly from Department of Aerospace Engineering, Mississippi State University, Dr. Ming Jiang from Livermore National Labs and Dr. Yootai Kim, Samsung Research, Korea for providing the code for CFD simulations and vortex extraction.

References

[1] Rakesh Agrawal, King-Ip Lin, Harpreet S. Sawhney, and Kyuseok Shim. Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases. In *Very Large Data Bases (VLDB) Conference*, 1995.

[2] James F. Allen. An interval-based representation of temporal knowledge. In *International Joint Conference on Artificial Intelligence*, 1981.

[3] S.K. Card, J.D. Mackinlay, and B. Schneiderman. *Information Visualization: Using Vision to Think*. Morgan Kauffman Publishers, 1999.

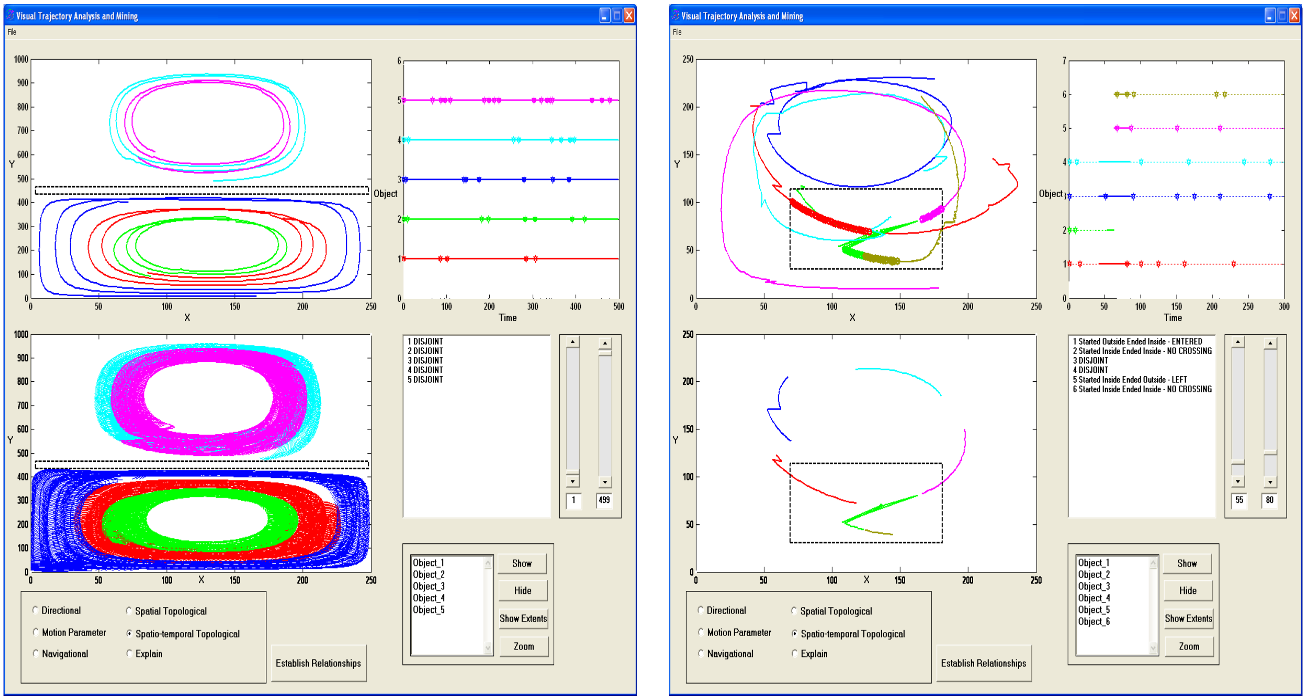

Figure 5: (a) Finding a region where no vortex entered (b) Explaining the critical event: Bifurcation

[4] Luca Chittaro and Carlo Combi. Representation of temporal intervals and relations: Information visualization aspects and their evaluation. In *International Symposium on Temporal Representation and Reasoning*, 2001.

[5] Max J. Egenhofer. Reasoning about binary topological relations. In *Symposium on Large Spatial Databases*, 1991.

[6] Kristin Eickhorst, Peggy Agouris, and Anthony Stefanidis. Modeling and comparing spatiotemporal events. In *DG.O National Conference on Digital Government Research*, 2004.

[7] Martin Erwig and Markus Schneider. Visual specification of spatio-temporal developments. In *IEEE Symposium on Visual Languages*, 1999.

[8] Martin Erwig and Markus Schneider. Spatio-temporal predicates. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):881–901, 2002.

[9] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. Fast Subsequence Matching in Time-Series Databases. In *SIGMOD Conference*, 1994.

[10] Ghassan Hamarneh and Tomas Gustavsson. Deformable spatio-temporal shape models: extending active shape models to 2d+time. *Image Vision Comput.*, 22(6):461–470, 2004.

[11] Harry Hochheiser and Ben Shneiderman. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004.

[12] Ming Jiang, Tat-Sang Choy, Sameep Mehta, Matt Coatney, Steve Barr, Kaden Hazzard, David Richie, Srinivasan Parthasarathy, Raghu Machiraju, David Thompson, John Wilkins, and Boyd Gatlin. Feature mining paradigms for scientific data. In *SIAM International Conference on Data Mining*, 2003.

[13] Ming Jiang, Raghu Machiraju, and David Thompson. Geometric verification of swirling features in flow fields. In *Proceedings of IEEE conference on Visualization*, 2002.

[14] Yootai Kim and Raghu Machiraju. Swirling Images. In *OSU-CISRC-1/06- TR03*, 2006.

[15] George Kollios, Dimitrios Gunopulos, and Vassilis J. Tsotras. On indexing mobile objects. In *Principles of database systems*, 1999.

[16] Flip Korn, H. V. Jagadish, and Christos Faloutsos. Efficiently supporting ad hoc queries in large datasets of time sequences. In *SIGMOD Conference* , 1997.

[17] Jessica Lin, Eamonn J. Keogh, and Stefano Lonardi. Visualizing and discovering non-trivial patterns in large time series databases. *Information Visualization*, 4(2):61–82, 2005.

[18] Sameep Mehta, Steve Barr, Alex Choy, Hui Yang, Srinivasan Parthasarathy, Raghu Machiraju, and John Wilkins. Dynamic classification of defect structures in molecular dynamics simulation data. In *SIAM International Conference on Data Mining*, 2005.

[19] Sameep Mehta, Raghu Machiraju, and Srinivasan Parthasarathy. Towards Object based Trajectory Representation and Analysis. In *OSU-CISRC-03/06- TR30*, 2006.

[20] Mohamed F. Mokbel, Thanaa M. Ghanem, and Walid G. Aref. Spatio-temporal Access Methods. *IEEE Data Engineering Bulletin*, 2003.

[21] Kin pong Chan and Ada Wai-Chee Fu. Efficient Time Series Matching by Wavelets. In *International Conference of Data Engineering*, 1999.

[22] C.R. Rao and S Suryawanshi. Statistical analysis of shape of objects based on landmark data. In *Proc Natl Acad Sci USA, 93(22)*, 1996.

[23] J. Roddick and B. Lees. Paradigms for spatial and spatio-temporal data mining, 2001.

[24] Simonas Saltenis, Christian S. Jensen, Scott T. Leutenegger, and Mario A. Lopez. Indexing the positions of continuously moving objects. *SIGMOD Rec.*, 29(2), 2000.

[25] Deborah Silver and Xin Wang. Tracking and visualizing turbulent 3d features. *IEEE Transactions on Visualization and Computer Graphics*, 3(2), 1997.

[26] Yufei Tao, Dimitris Papadias, and Jimeng Sun. The tpr*-tree: An optimized spatio-temporal access method for predictive queries. In *Very Large Database Conference*, 2003.

[27] Hui Yang, Srinivasan Parthasarathy, and Sameep Mehta. A generalized framework for mining spatio-temporal patterns in scientific data. In *Knowledge Discovery and Data Mining*, 2005.

[28] Hui Yang, Srinivasan Parthasarathy, and Sameep Mehta. Towards association-based spatio-temporal reasoning. In *IJCAI: Workshop on Spatio-temporal Reasoning*, 2005.