# Visual Exploration of Spatio-temporal Relationships for Scientific Data

Sameep Mehta*
Computer Science and Engineering
The Ohio State University

Srinivasan Parthasarathy
Computer Science and Engineering
The Ohio State University

Raghu Machiraju
Computer Science and Engineering
The Ohio State University

**ABSTRACT**

Spatio-temporal relationships among moving objects, if derived in a meaningful fashion, can provide useful information about the evolution of an individual object and its interactions with other objects. However, extracting such useful relationships without user guidance is a cumbersome and error prone process. In this paper, we present a visual analysis system which interactively discovers such relationships from the trajectories of the moving objects. We describe analysis algorithms to derive various spatial and spatio-temporal relationships. We, then, present a visual interface through which the user can interactively select spatial and temporal extents to guide the process of knowledge discovery. We show the usefulness of our proposed algorithms on the 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.

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

## 1 INTRODUCTION

In this paper we describe a visual reasoning and knowledge mining system to understand the spatial and spatio-temporal relationships[1] among evolving features[2]. In this work, we focus on features stemming from scientific simulations. We describe an interactive visual interface coupled with a strong analysis component which helps the user to derive information about the evolution of a single feature and complex relationships among different features.

A fundamental property of spatio-temporal features is that, *the spatial positions of the features change over time*. This change in positions can be characterized by motion parameters including linear velocity and angular velocity. However, in scientific datasets, the extent and the shape of the objects also change frequently and are important to describe the evolution of the object completely. The changes in these spatial properties can lead to interesting phenomena like dissipation and creation of new features. For example, a shrinking object may cease to exist at some later time instant. We can capture the change in the size of the object by using scale parameters. The evolution of the individual features can also induce changes in the relationships among features e.g. two far apart features moving towards each other will most likely interact at some future time instant. These interactions can result in the occurrence of the critical events including merging and bifurcation [26]. Therefore, to extract useful and important information from the evolving features, it is necessary that the spatial and spatio-temporal relationships amongst them are correctly identified. Efficient mining of these relationships forms the major component of the analysis algorithms of our system. To accomplish this task we use the following attributes to represent the trajectory of an object: *i) positions, ii) change in positions over time, iii) extent and iv) change in extent over time.*

Another important characteristic of the objects is that most of the time they interact exclusively with other objects present in the neighboring spatial region(s). Identification of such a region can help in establishing relationships among the objects. There are infinite possibilities for selecting the size and position of such region(s). Choosing a large area will result in deriving non-existent relationships. Similarly, choosing a small area will result in missing some important interactions. The size and position of this neighborhood differs not only across datasets but also varies across different objects in the same dataset. This problem becomes more complicated when the objects are moving and changing extents. In such cases, one single method to define the neighborhood for all the objects can produce suboptimal or even wrong results. Similarly, determining a useful time interval for analysis poses yet another challenge to the whole 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 object requires individual attention for selecting a meaningful region and time interval for detailed analysis. To accomplish this task there is strong need for a visual interface though which the user can interactively select the extents. Moreover, often the trajectories of different objects overlaps, therefore, the user also needs 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 also plays a very important role in the whole analysis process. However, displaying the extents for all the objects for the whole time span of the simulation will result in a highly cluttered visualization. It will be extremely difficult to glean any interesting information from such a cluttered view. Therefore, the extents should be displayed only on the user's request. These features together with the interactive selections form important visualization components in our system.

Both, the analysis and visualization components help users discover useful information from the datasets efficiently by making the search process and analytical reasoning more focused and goal driven. The visualization component enables the user to interactively select interesting spatial and temporal extents to perform analysis on. The analysis results provide users with the useful information about the behavior of objects. These results are displayed using the visualization component. The user uses the displayed results and feature like zoom and filter to refine the extents. This process is repeated till the user discovers the information he or she is seeking or finds some 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. Additionally, we also provide support for the *zoom, filter and details on demand* paradigm [3].

2. We present algorithms for automatically deriving various spa-

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

[1]Please note that if the time interval is ignored or a single time instant is considered, then the derived relationships are purely spatial

[2]Features are defined as region of interest in scientific datasets. In this article we use the terms feature and object interchangeably.

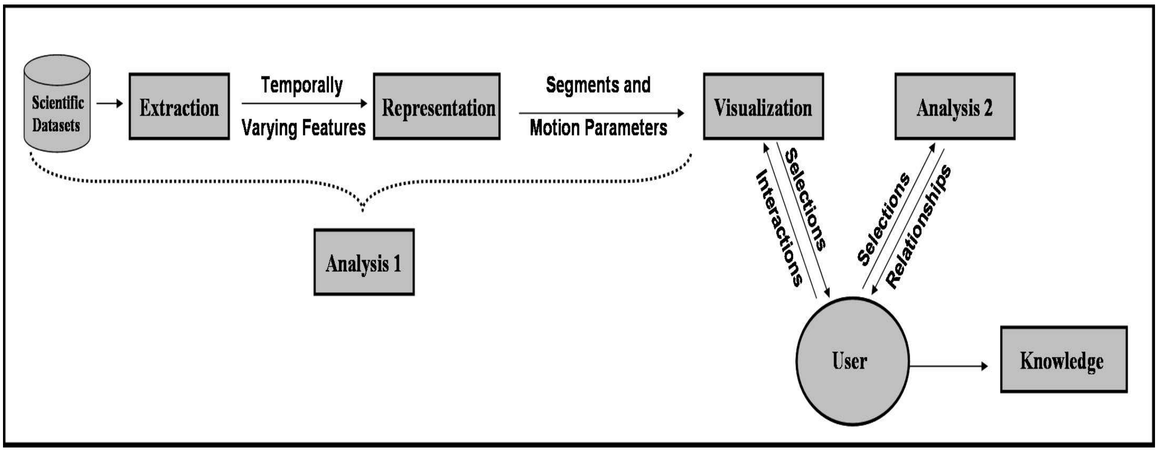



Figure 1: Overview of the System for Understanding Trajectories of Scientific Objects

tial and spatio-temporal relationships like the topological relationships proposed by Egenhofer [5].

3. We empirically demonstrate the usefulness of our algorithms on datasets originating from computational fluid dynamics. We also discuss the use of domain knowledge coupled in our algorithms for extracting useful information from these trajectories.

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

## 2 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 linear velocity $\vec{v}$ and angular velocity $\vec{\omega}$ are estimated [20]. Apart from being physically intuitive and meaningful, this representation also reduces memory overheads. Moreover, important characteristics about the motion can also be ascertained by investigating 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 then used by 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.

- **Analysis2 - Deriving Relationships:** This component acts as the backend engine for finding the relationships. In this paper, we focus on directional, navigational and topological relationships. Once the high level relationships are established the user can further refine them by refining the spatial and temporal extents. The analysis algorithms can also help explain the likely cause of critical events like bifurcation and merging.

The user iterates through component 2 and 3, reducing the search space in each iteration to obtain important and meaningful information about the object's behavior.

Our previous research efforts have largely focused on the first component of the system [14, 13, 19]. Therefore, in this paper, 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. [13] 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 used the algorithms presented in [13, 14] for detecting vortices from temporally varying datasets.

Recently, we proposed a parametric scheme for representing the motion of evolving features [20]. 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**). We used least square minimization to estimate the parameters between every two consecutive frames in the dataset. Next, we employed a clustering algorithm to segment the trajectories into piecewise smooth subtrajectories. The clustering algorithm uses the estimated MPV as a feature vector. We used weighted euclidean distance to compute the distance between two feature vectors. Each subtrajectory is represented by a single MPV. We strongly believe that this representation is physically intuitive and 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 [20] no visual component was presented. The focus there was to motivate the need and evaluation of this representation. Most of the analysis reported in [20] was performed by trying several spatial and temporal extents.*

*The cumbersome manual process motivated us to develop this visual interface.*

## 3 ALGORITHM

In this section we present last two components of the above mentioned system in details. We first present the basic notation used throughout the paper.

**Basic Notation:** $S$ denotes a time varying dataset with $N$ steps monitoring the movement of $n$ objects $O = \{O_1, O_2, \ldots, O_n\}$ An object $O_r$ is represented by $K$ points (landmarks) sampled from the surface of $O$ [29, 23]. 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 $\delta$. After determining motion parameter vectors **(MPV)** and segmentation, the $j^{th}$ sub-trajectory of $O_r$ is denoted by $O^p_{r,j}$ 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 $\{P\}$ represents the MPV of the $j^{th}$ sub-trajectory of $O_r$. The time interval of the $j^{th}$ segment is $[t^j_1, t^j_2]$. The stored $\{x, y, z\}$ points are landmarks describing the shape of the object at start of the segment i.e at time $t^j_1$. Description of the object anywhere else in $j^{th}$ segment can be obtained by recursively applying the parameters to the shape descriptors.

Next, we describe the main analysis tasks and relationships we are interested in. *Please note that even though the visualization component precedes the analysis component in Figure 1, we present the analysis first so that the motivation behind the design of visual component and user interactions can be explained clearly.*

### 3.1 Analysis2 - Deriving Relationships

The set of $n$ objects is denoted by $O = \{O_1, O_2, \ldots, O_n\}$. The time interval 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 to identify the connection between $R$ 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*) $R$ iff the whole object is enclosed (not included) in $R$. The *overlap* is defined if some part of $O_{r,l}$ is outside $R$. 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 $true$ is point $p$ lies side $R$, $false$ 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:

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

  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 four events *enter, leave, disjoint and cross*. $O_r$ 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. However, to distinguish *disjoint* and *cross* events we need process very time step because $O_{r,s}$ and $O_{r,e}$ will be outside $R$, in both the cases. Finally, we derive these relations by using the above described *inside, outside and overlap* operators as follows:

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

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

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

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

- **Navigational Relationships -** This analysis is used to understand the motion characteristics of the objects. First the user can select a spatial region and/or time interval and all the motion parameters can be displayed. Next, we use the *follows* operator as described by Roddick et.al [24]. Given a spatial region $R$ (and/or time interval), this analysis finds if within $R$ a trajectory demonstrates similar motion to itself or to other trajectories. In our representation, we find the motion parameters for each connected trajectory. Next, the distance between these parameters is calculated. If the distance $\leq user\ defined\ threshold$, 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 can also be viewed in form of a animation, providing more details about the behavior of the object at each time step. We discuss this aspect in detail in Section 4.

### 3.2 Visualization - User Interface

Even though the analysis component seems 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, the answer to this problem requires a brute force algorithm. For example, assume that *we want to find 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. Such an $R$ can be found by performing exhaustive search over the whole space changing the size, orientation and position of $R$ each time. This process is computationally prohibitive. However, with visual interfaces the user can start by defining a coarse region first and refining it by changing the size and orientation to find an appropriate $R$. Therefore, the user can identify potential regions very quickly thereby making the search process more focused, efficient and meaningful.

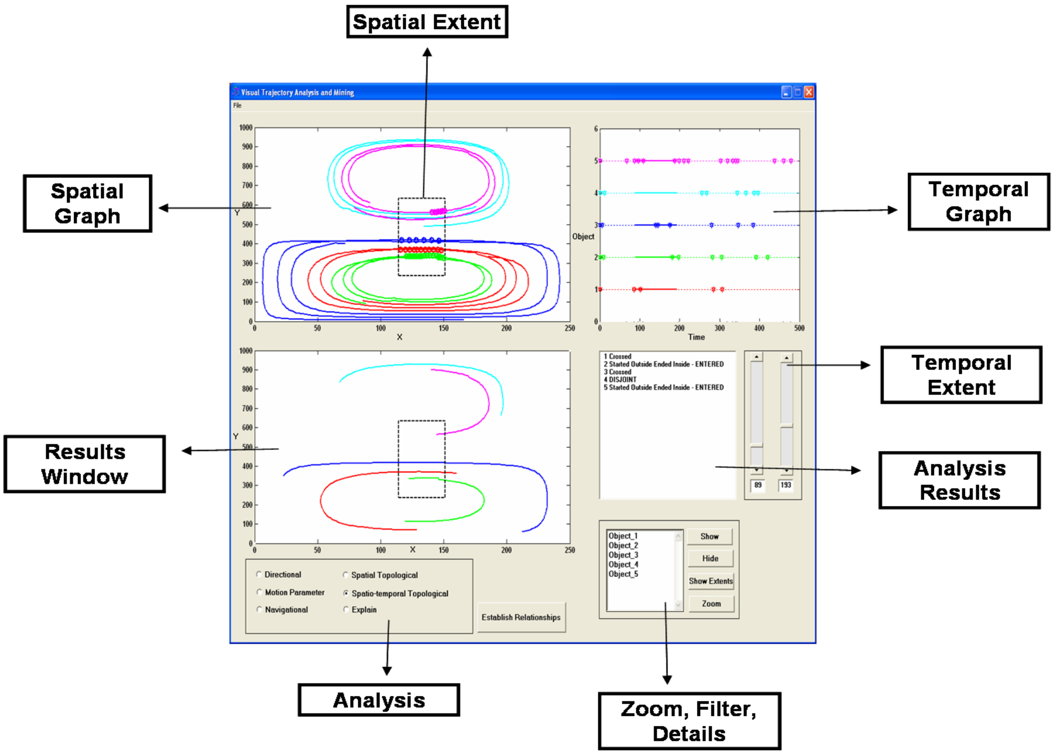



Figure 2: Overview of the Visual Interface

In this section we describe the visual representation we use. Specifically, we present two graphs i) spatial graphs (SG) 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 details and also point to the use of these graphs for visual analysis and reasoning.

- **Spatial Graphs (SG):** This graph displays the trajectories in *xyz* 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 object.

- **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 subtrajectory.

- **User Interactions:** Please recall that for most of analysis tasks we need a spatial region $R$ and a time interval $[t_s, t_e]$. The user can interactively select 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 ability to hide some of the trajectories to focus on the visually more interesting trajectories. Once the user is satisfied with spatio-temporal extents, he or she can start the analysis part by invoking function calls to the back-end engine. The results of the analysis are presented to the user. The results can be displayed either statically or as an animation. Based on the results, the user can refine the search space and again use analysis tools. This iterative process is continued until the final desired information is extracted. The user can not only iterate through 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 towards each other. Spatial proximity can be ascertained by directional relationships and the direction of the movement can be found using navigational relationships.

Now we present an overview of our visual interface, highlighting the use of major parts of the interface. Figure 2 show one snapshot of our 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 for establishing correspondence. The markers on $TG$ indicate the segment boundaries. *Please note that the length of the interval between some markers 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 can be 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 (Analysis) shows all the operations which our system currently supports. The Result Window ($RW$) visually displays the result of analysis. These results, along with more information, are displayed in text format in Analysis Result ($AR$) window.

## 4 RESULTS

In this section we demonstrate the use of our system on datasets originating from computational fluid dynamics. We used the simulation by Kim and Machiraju [15] to generate the datasets. The features (vortices) are detected by using the algorithms proposed by Jiang et.al [14]. 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

- **Spatio-Temporal Topological Relationships-** Figure 2 is an example of spatio-temporal topological analysis. The *RW* (Result Window) displays the parts of trajectory which are active during the selected time interval. In the *SG* (Spatial Graph) 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 object 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, we only make use of first slider. 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 vortices which are inside R. 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.*

- **Spatial Topological Relationships-** Figure 3(b) shows the derived spatial topological relationships. As in the last example, only first slider is used. *RW* shows the selected *R* and the position of vortices 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*.

- **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 instance in 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. Therefore, even though these relationships are derived very efficiently, they can sometimes provide incomplete results, e.g. object 2 started *inside R* and ended *inside R*, topological analysis will return *no crossing* as the answer. However, by using the explain feature we can easily determine that the object moved *outside* at $t = 218$ and moved back in at $t = 263$. The explain 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 obtain more information about the interesting objects. The information can then be used to construct temporal rules. The rules generated for objects 1 and 2 are:

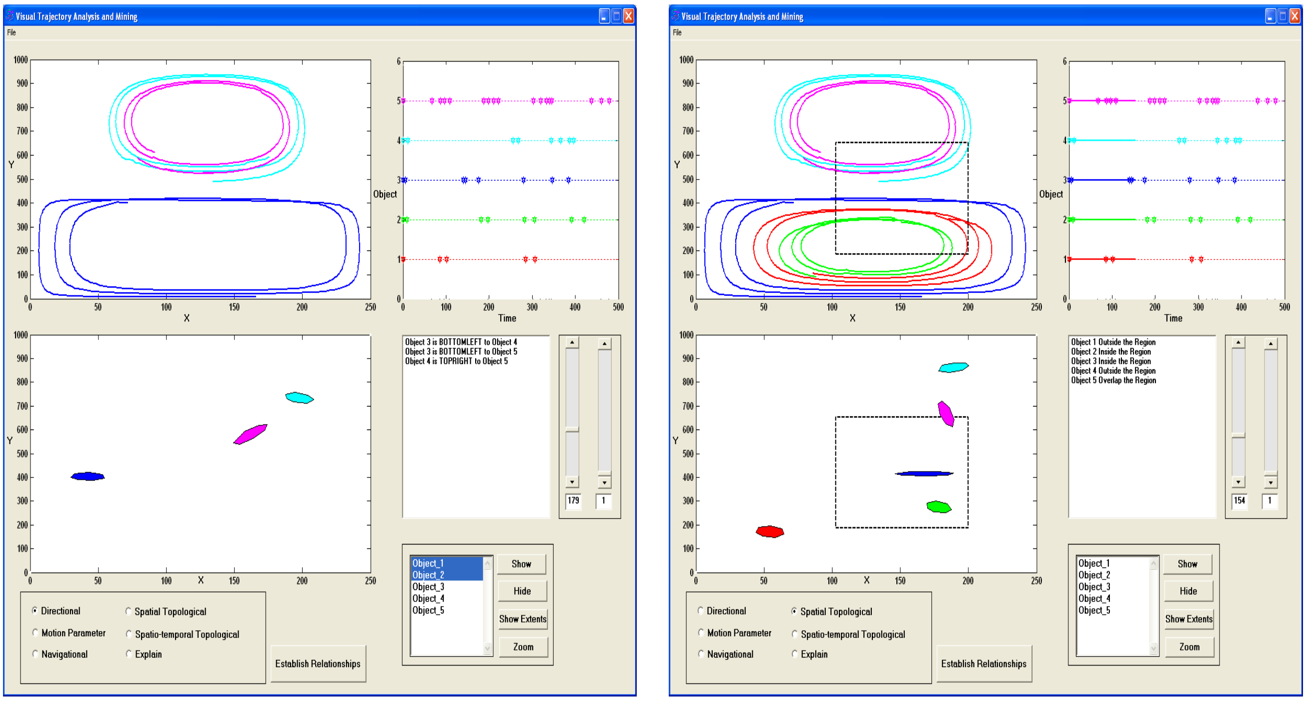
  Object 1 is *Inside* between $[200,215]$
  Object 1 is *Outside* between $[216,262]$
  Object 2 is *Inside* between $[200,218]$
  Object 2 is *Outside* between $[219,262]$
  Object 2 is *Inside* between $[263,272]$

  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. Only spatial region is needed to find these relationships. *RW* shows the zoomed view of the trajectories in the selected spatial region. *AR* displays the final relationships. 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 object 4 is hidden. Please note that *follows* is a symmetric relation between two objects. Therefore, we only display it once.

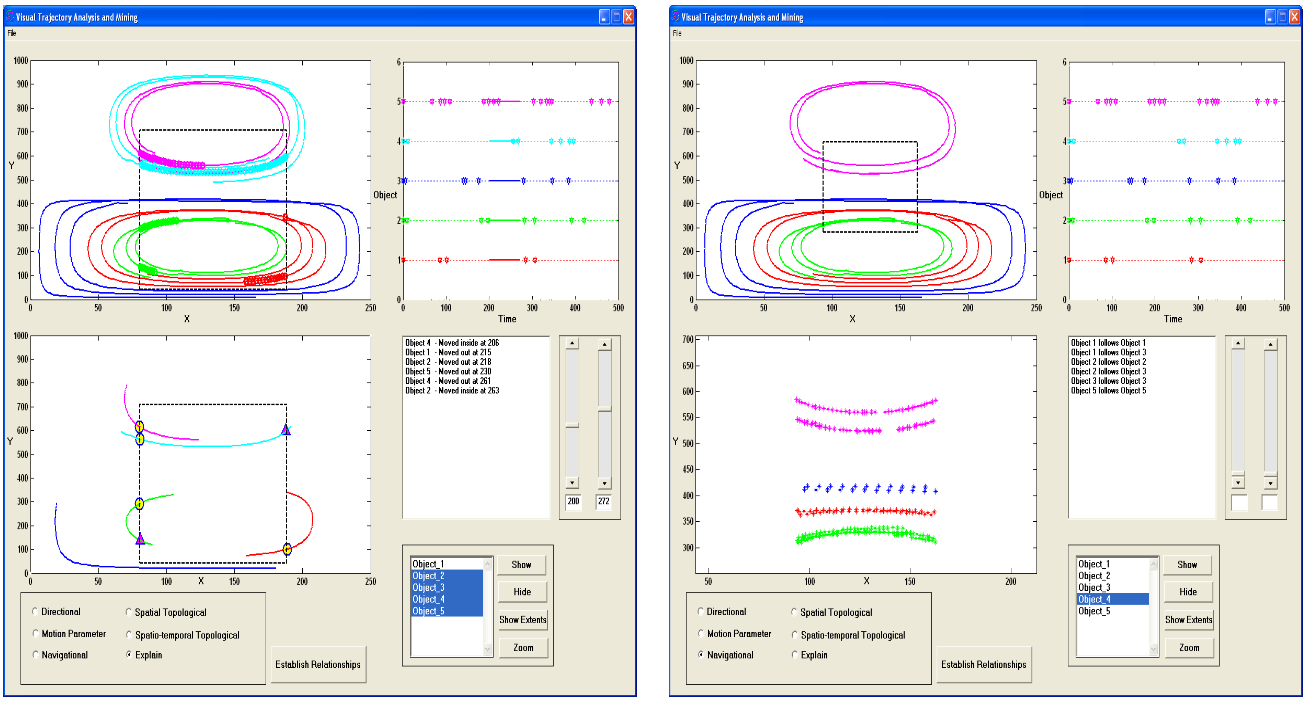
- **Discovering Interesting Spatial Regions-** Our system can be used to interactively discover interesting regions in the dataset. We present one such example in Figure 5(a). The goal here was to find the largest spatial region *R* such that no vortex was present in *R* given a time interval. If the interval covers the entire span of the simulation, then presence of such an area suggests that the initial simulation parameters does not allow vortices to enter this area. Figure 5(b) 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 was overlapping *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 $t = 75$. We used our system to establish that object 2 bifurcates into 5 and 6. Next, we tried to understand the the process which is most likely responsible for 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 $[6875]$. During the interval $[71,74]$, the shape of the object 2 was deformed in such a fashion that it cannot be



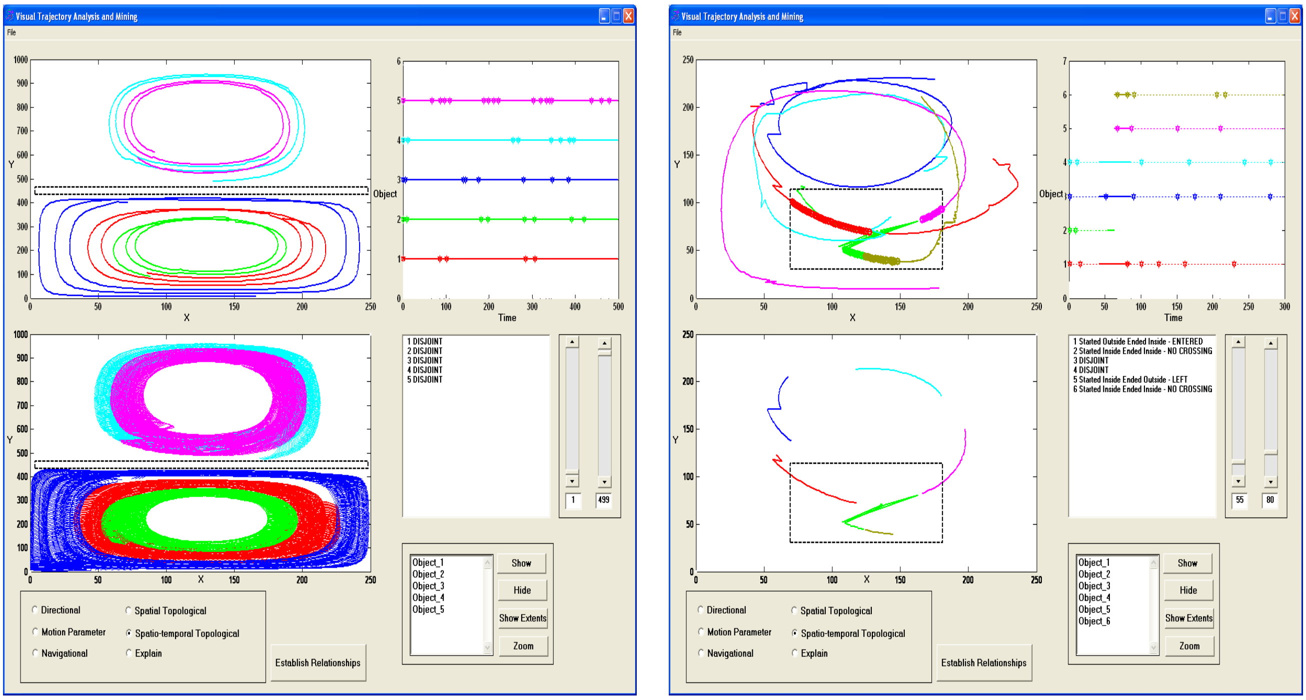

(a) (b)

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




(a) (b)

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




(a) (b)

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

correctly 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.

## 5 Related Work

Our motion estimation algorithm is closely related to trajectory representation present in existing literature in data mining and databases. These algorithms can be divided into two broad categories i) native (*xyz*) space representation and ii) parametric space representation. Please see [21] for an excellent survey on most popular native space representation techniques. Techniques which exploits one dimensional time series including those based on DFT [1, 9], DWT [22], SVD [17] are not directly applicable in this context. Kollios et.al [16], Saltenis et.al. [25] and Tao et.al [28] represented the trajectories by subsegments where each sub-segment has constant linear velocity. The usefulness of storing linear velocity instead of actual object location is shown by efficient query processing and the low overhead in terms of updates to the index structure. All the approaches reviewed so far abstract the object by a point (typically center of mass). This simplification, even though, produces highly efficient algorithms misses crucial information. For example, 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 a object at two successive time instants, a translation matrix (and hence 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.

Allen [2] proposed temporal interval algebra. The author described 13 basic relationships which can hold among different time intervals. Few example of such relationships are *During, After, Before*. Egenhofer [5] presented 9 intersection models 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 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, the authors also presented some guidelines on the representation of a sequence of spatio-temporal predicates [7]. Recently, we [29, 30] 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 [12] presented a tool TIMESEARCHER for visualizing and interactively querying time series datasets. Recently Lin et.al. [18] 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 only temporal data. Hamarneh and Gustavsson [10] presented algorithms for modeling and segmenting 2d time varying shapes. Even though the segmentation algorithm is similar in some ways to our clustering algorithm but no explicit modeling of motion parameters is done. Eickhorst et.al [6] proposed spatio-temporal helix to model the trajectory of the object. The authors showed the use of this representation for comparing two trajectories. However, visualization and analyzing relationships among objects was not the focus of [10, 6]. Hao et.al [11] proposed methods for visualizing large scale time varying molecular datasets. Finally, Stockinger et.al [27] demonstrated the use of bit map indexing for interactive querying and visualization for large scientific datasets.

## 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.

## 7 Acknowledgments

This work is funded by the following NSF grants NGS-0326386, ACI- 0234273 and NSF 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 Yootai Kim, Department of Computer Science and Engineering, Ohio State University for providing 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.

[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] X. Hao and A. Varshney. Real-time visualization of large time-varying molecules. In *Proceedings of the High-Performance Computing Symposium '04*, April 18 - 22, 2004.

[12] 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.

[13] 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.

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

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

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

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

[18] 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.

[19] 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.

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

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

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

[23] 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.

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

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

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

[27] Kurt Stockinger, John Shalf, Kesheng Wu, and E. Wes Bethel. Query-driven visualization of large data sets. In *IEEE Visualization*, page 22, 2005.

[28] 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.

[29] 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.

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