

Geometric Querying of Time-Dependent Data for Data Mining in Molecular Dynamics

Olga Sourina and Nikolay Korolev
Nanyang Technological University, Singapore
eosourina@ntu.edu.sg

Abstract

Temporal Databases and Data Warehouses are essential components of intelligent information systems in Cyberworlds. This paper describes a geometric model for querying time-dependent data in databases and warehouses. An implementation and application of the model for querying of results of numerical simulation in Molecular Dynamics is discussed. Data are interpreted geometrically as multidimensional points with time dimension. A geometric query is a query solid of any shape specified by its parameters, location and time. These queries are formulated with geometric objects and operations over them to form the query solid. The geometric objects and operations are described with implicit functions. With the uniform geometric model for querying time-dependent data, 3D visualization tools can be naturally incorporated into the molecular dynamics visualization system to pose the queries.

1. Introduction

Querying of time-dependent data is a classical problem in spatio-temporal, and temporal databases. Research in this area has become even more important because of recent fast development of data warehouses. Nowadays, amounts of data collected are rapidly increasing, and data warehousing becomes an important strategy to integrate heterogeneous information sources in collaborative environment in Cyberworlds.

Usually we are interested both in the data and in the underlying processes and structures that these data provides. In case of time-dependent data, we often are interested in forecasting future developments based on trends and cyclic behaviors in the data [1]. For instance, forecasting the consumer shopping behavior would give companies strategic advantages. Predicting behaviors of molecules in DNA could also provide a new insight in the molecular dynamics. However, finding such valuable information hidden in data is a complicated task. Data mining denotes the approach to analyze this data and to extract valuable information to explain processes and to make decisions.

The essential component of data analysis in databases and data warehouses is querying of data. The user of

temporal databases and warehousing systems often needs to be interactively involved in the process of querying. It becomes more and more important for the modern databases and warehouses to give the user an easy understanding of both the data set and the query results. Visualization offers the user an intuitive way of analysis that can help to discover data patterns and structures and query the data [2]. Data visualization techniques, when incorporated with querying algorithms, could improve interpretability and usability of the time-dependent data and analysis of that data. First, the user is interested not only in the results of querying, but also in the querying process and spatial relationship between data changing over time. Therefore, the user should be involved into the querying process to make it more efficient and accurate. Thus, visualization techniques could be used not only for visualization of results of querying and data mining but for visualization of querying process. The user should also have an opportunity to select the projection directions for high dimension data set. To incorporate visualization techniques, the existing database systems use the result of querying as the input for visualization system that is costly and inefficient. The better solution is to combine two processes together, which means to use the same model to query and visualize time-dependent data.

As it was mentioned above, querying of time-dependent data is a classical problem in databases and warehousing. The goal of works in this area is to propose data representation model and query model able to handle time-dependent geometries including those changing continuously that describe moving objects [3, 4]. Spatio-temporal predicates are introduced to query time-dependent data [4].

In work [5], we proposed and fully described geometric query model with implicit functions [6]. Then, in work [7], we proposed a uniform geometric model for clustering and querying multidimensional data. In this paper, we extend the uniform geometric query model to handling time-dependent data. In our novel model, we proposed to use implicit function in spatio-temporal predicate implementation. This allows us to pose complex shape queries changing over the time. Our extended model allows us to integrate visualization and querying of data in spatio-temporal, temporal databases and/or data

warehouses. Temporal query languages are not discussed in this paper. Based on the proposed geometric query model we developed graphical user interface to pose time dependent complex shape queries on time-dependent data.

The paper is organized as follows. In Section 2, the geometric model for querying time-dependent data is introduced. Implementation of the query model as geometric query system is described in Section 3. Application of the proposed querying model for solving problems in molecular dynamics is discussed in Section 4. In Section 5, conclusion and future work are presented.

2. Query Model for Time-Dependent Data

Let us introduce the formal mathematical specification of the geometric query model for time-dependent data with the function-based representation of geometric solids. We extended the geometric query model proposed in [5] with time dimension. A geometric object can be a set of points $P = \{[x_1, x_2, \dots, x_n, t]\} = \{[X, t]\}$ in n dimensional Euclidean space E^n , and t is time. As it was proposed in work [5], primitive solid objects are defined with implicit functions as $f(x_1, x_2, \dots, x_n) \geq 0$ in Euclidean space E^n . The implicit function $f(x_1, x_2, \dots, x_n) \geq 0$ can be defined analytically or by procedure. Such functions define closed n -dimensional objects in E^n space under the following conditions:

- $f(\mathbf{X}) > 0$ - for the points inside the object,
- $f(\mathbf{X}) = 0$ - for the points on the object boundary,
- $f(\mathbf{X}) < 0$ - for the points outside the object,

In our model, query solid can have time-dependent parameters and/or coordinates that can be defined analytically or by procedure. Thus, the geometric query model consists of the following geometric objects:

- n -dimensional points $P = \{[x_1, x_2, \dots, x_n, t]\}$ where t is time;
- time-dependent 3-dimensional primitive geometric objects for the construction of a query solid using geometric operations.

The following is an implicit function representation of the primitive time-dependent 3-dimensional geometric solids that could be used for construction of geometric criteria:

Halfspace:

$$G_1: f_1(\mathbf{X}, t) = f_1(x_1, x_2, x_3, t) = (x_1 - a[t]) \geq 0$$

Where a is some real number ($a \in \mathbb{R}$)

Sphere:

$$G_1: f_1(\mathbf{X}, t) = r[t]^2 - (x_1 - x_{0,1}[t])^2 - (x_2 - x_{0,2}[t])^2 - (x_3 - x_{0,3}[t])^2 \geq 0$$

Where $x_{0,1}, x_{0,2}, x_{0,3} \in \mathbb{R}$

Ellipsoid:

$$G_1: f_1(\mathbf{X}, t) = 1 - ((x_1 - x_{0,1}[t])/a_1[t])^2 - ((x_2 - x_{0,2}[t])/a_2[t])^2 - ((x_3 - x_{0,3}[t])/a_3[t])^2 \geq 0$$

where $x_{0,1}, x_{0,2}, x_{0,3} \in \mathbb{R}$ and $a_1, a_2, a_3 \in \mathbb{R}$.

Cone:

$$G_1: f_1(\mathbf{X}, t) = ((x_1 - x_{0,1}[t])/a_1[t])^2 - ((x_2 - x_{0,2}[t])/a_2[t])^2 - ((x_3 - x_{0,3}[t])/a_3[t])^2 \geq 0$$

where $x_{0,1}, x_{0,2}, x_{0,3} \in \mathbb{R}$ and $a_1, a_2, a_3 \in \mathbb{R}$.

Cylinder:

$$G_1: f_1(\mathbf{X}, t) = ((x_1 - x_{0,1}[t])/a_1[t])^2 - ((x_2 - x_{0,2}[t])/a_2[t])^2 \geq 0$$

Where $x_{0,1}, x_{0,2} \in \mathbb{R}$ and $a_1, a_2 \in \mathbb{R}$.

By further declaring that our model is open to any type of objects that can be defined implicitly with some functions $f(x_1, x_2, x_3, t) \geq 0$, we could avoid the problem of a minimum set of primitives and to change this set depending on the application problem to be solved.

Geometric operations are applied to primitive geometric objects to obtain complex geometric shapes at each time point. The analytical definition of set-theoretic operations is realized in the form proposed by Ricci [8], where operations over implicit functions are considered. Affine transformations (translation, rotation and scaling) are also used to increase an expressive power of the proposed geometric model. Geometric operations include set-theoretic union, intersection, difference and orthographic projection.

Mathematically,

Union: $G_3 = G_1 \cup G_2$ of two objects $G_1 \subset E^n$ and $G_2 \subset E^n$ with the descriptive functions f_1 and f_2 will be defined as $f_3 = f_1 \vee f_2 = \max(f_1, f_2) \geq 0$, where $G_3 \subset E^n$.

Intersection: $G_3 = G_1 \cap G_2$ of two objects $G_1 \subset E^n$ and $G_2 \subset E^n$ with the descriptive functions f_1 and f_2 will be defined as $f_3 = f_1 \wedge f_2 = \min(f_1, f_2) \geq 0$, where $G_3 \subset E^n$.

Complement: $G_2 = \neg G_1$ of object $G_1 \subset E^n$ with the descriptive functions f_1 will be defined as $f_2 = -f_1 \geq 0$.

Difference: $G_3 = G_1 \setminus G_2$ between objects $G_1 \subset E^n$ and $G_2 \subset E^n$ with descriptive functions f_1 and f_2 will be defined as $f_3 = f_1 \wedge (-f_2) = \min(f_1, -f_2) \geq 0$, where $G_3 \subset E^n$.

Translation: $G_2 = T(G_1)$ of object $G_1 \subset E^k$ with descriptive functions f_1 by a_1, a_2, \dots, a_n will be defined as $f_1(x_1 - a_1, x_2 - a_2, \dots, x_n - a_n) \geq 0$.

Rotation: $G_2 = R(G_1)$ of object $G_1 \subset E^k$ with descriptive functions f_1 of angle α about some axis will be defined as $f_1(x'_1, x'_2, \dots, x'_n) \geq 0$ where $[x'_1, x'_2, \dots, x'_n, 1] = R^{-1}[x_1, x_2, \dots, x_n, 1]$ and R^{-1} is an inverse matrix of rotation.

Scaling: $G_2 = S(G_1)$ of object $G_1 \subset E^k$ with descriptive functions f_1 in s_1, s_2, \dots, s_n times will be defined as $f_1(x_1/s_1, x_2/s_2, \dots, x_n/s_n) \geq 0$.

We introduce a point/solid predicate for query implementation. Let P be a point in Euclidean space E^n and t is time, G_1 be a query solid described with implicit function f_1 defined with time-dependent parameters and location changing over time, bG_1 be a boundary of G_1 and

iG_1 be an interior of G_1 . Then a point/solid predicate is described with the implicit function representation of the geometric object G_i by a 3-valued predicate:

$$S_3(P, G_1) = \begin{cases} 0, & \text{if } f_1(x_1, x_2, \dots, x_n, t) < 0 \quad P \notin G_1 \\ 1, & \text{if } f_1(x_1, x_2, \dots, x_n, t) = 0 \quad P \in bG_1 \\ 2, & \text{if } f_1(x_1, x_2, \dots, x_n, t) > 0 \quad P \in iG_1 \end{cases}$$

3. Implementation and Results

To visualize and query multidimensional temporal data we implemented the graphical user interface GUI. The proposed geometric model for querying time-dependent data is meant to be the user's model as well as the formal foundation for the implementation of the visual querying of time-dependent data. The points mapped from the temporal database and/or warehouse, the reconstructed geometric solids, the query formulation, and the resulting geometric objects changing over time are visualized. To get an initial impression of data, the data are visualized as 3-dimensional snapshots of point clouds at each time point. After the data is visualized as 3-dimensional projection at some time point, a complex geometric query solid with union, intersection or other operations over primitive geometric solids is posed by the user. These primitive query solids currently are cuboid (box), ellipsoid, cone, and cylinder. In addition, any point of clouds can be located and identified in the database/warehouse. In Figure 1, an example of querying of time-dependent data is shown. First, 3-dimensional projection of multidimensional points is mapped from the database and visualized as point clouds. A blobby solid can be reconstructed at each time point to show shape of point clouds changing over time. Then, a solid query is posed and time interval is chosen. Here, a query solid is a cylinder that does not change its parameters and location over time interval. The result of the query is time-dependent data and is visualized as set of snapshots or as file with animation.

With the proposed query model, the user could specify a query solid for each time point defining time-dependent primitive solids parameters and location analytically or by procedure. Currently, with the implemented GUI, the user constructs the query shape that does not change over time interval.

Geometric objects can be drawn opaque or transparent. We employ visualization techniques and advanced computer graphics algorithms for the implementation of the user interface. For better immersion stereo visualization is implemented. The system is implemented with the software Visualization Toolkit (VTK) where visualization is implemented with marching cube algorithm [9].

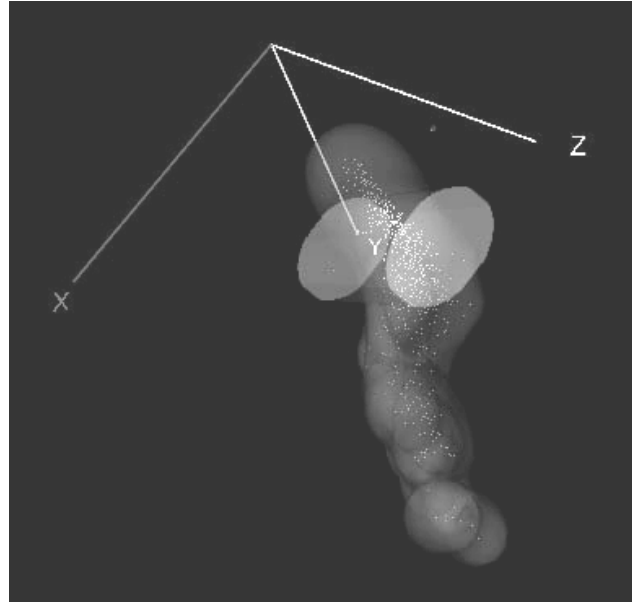


Figure 1. Querying of Dataset 1

4. Application of Geometric Querying for Solving the Problems of Molecular Dynamics

Let us apply our extended geometric query system for querying results of numerical simulation in Molecular Dynamics. We work with the resulting data of numerical simulation studying physics of DNA in interaction with solvent (water) different ions and various proteins. In this model, various rules and conditions are specified such as strength of bonding forces, charges of ions and other factors that affect the simulation. With this numerical model, we are able to obtain data changing over time that we use as the input for our geometric query system.

The time-dependent data consists of atom coordinates changing over time. Values are calculated at typically equal time intervals. Here, we use a common molecular dynamics data format as an input and output for our geometric query system. The file example is shown in Figure 2.

The number in the first line indicates the number of atoms and molecules. The second line specifies the snapshot time point. Other lines start with chemical symbols of the atoms or molecules followed by x , y and z coordinates of the atoms. There are many snapshots of data corresponding time sequence. In this work, we do not discuss time-dependent data representation in database/warehouse.

5662			
after	4760050. fs		
O5*	17.9594	18.3627	-15.0381
C5*	18.6039	19.6368	-14.7906
H5*1	19.5205	19.7209	-15.3361
H5*2	18.9048	19.6878	-13.7142
C4*	17.8721	20.8832	-15.2925
H4*	18.5781	21.6951	-15.4226
O4*	17.1124	20.6732	-16.5126
C1*	15.7985	21.2096	-16.4406
H1*	15.4994	21.9514	-17.2903
N9	14.7788	20.1067	-16.5919
C8	15.0443	18.8150	-16.6745
H8	15.9641	18.3845	-16.3364
N7	14.0859	18.0414	-17.0365
C5	12.9754	18.8711	-17.1513
C6	11.6840	18.6325	-17.6425
N6	11.2422	17.3738	-17.7645
HN61	10.2690	17.1461	-17.9857
HN62	11.9397	16.6761	-17.8173
N1	10.9699	19.7606	-17.7713
C2	11.4583	20.9850	-17.5668

Figure 2. Example of input and output data file

Besides geometric query system, we also use *gOpenMol* system to visualize results of the query. *gOpenMol* is a tool for the visualization and analysis of molecular structures combined with several applications for data analysis and presentation originated from quantum mechanics, Molecular Dynamics and other computational chemistry calculations. The system can be downloaded from the Internet (<http://www.csc.fi/gopenmol/>).

After studying the problems of molecular dynamics, we proposed the following types of queries that can be easily implemented with our geometric query system. Let us describe some of the queries.

Query 1. Find and display trajectories of atoms by atom name or by its location (x, y, z)

The user will be able to pick specific atoms that he or she is interested in, by their name or by location, and we will display only those selected atoms changing its location over time. For both parts, the number of atoms that could be chosen would be limited due to the large amount of data that the system may go through. Our tests were done with 500 files of 40 MB each.

Query 2. Find and display trajectory of atoms in the selected region where the region can be the final query solid constructed as a result of operations over primitive solids.

The region can be of various shapes such as a cube, an ellipse, a cylinder, a sphere or even a cone. The result of union, intersection, and/or subtraction operation over the primitive solids can be a query solid as well.

Query 3. Find trajectories of atoms for a specified time interval $[t_1, t_2]$

We can combine all three queries. The developed geometric query tools allow the user to formulate spatio-temporal queries that cannot be implemented directly using the selection operation of relational algebra. This will allow the user to choose specific atoms and at the same time an interval and a region in which the atoms moves over time. This will be particularly useful for observing when the chosen atoms move out of the region specified along their trajectory. In Figure 3, a snapshot of the file at time point is visualized. Then the solid query as the difference between two cylinders is posed. The result of the query at one time point is shown in Figure 4. The result of the query can be also visualized with the system *gOpenMol*.

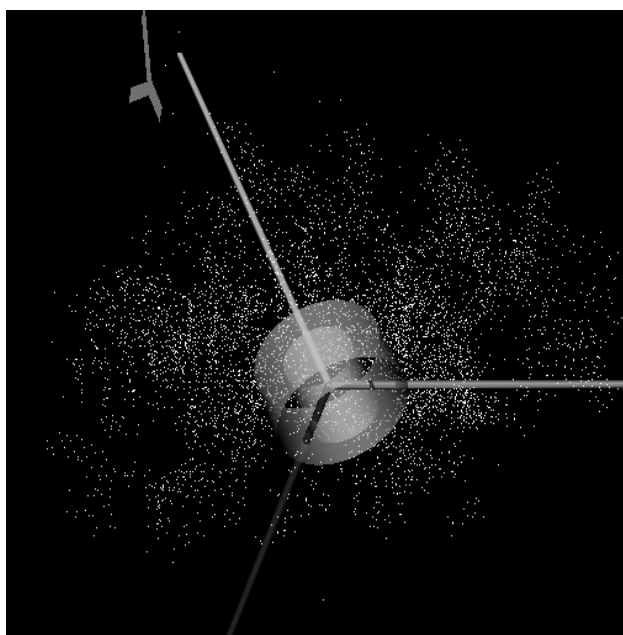


Figure 3. Querying of time-dependent data

The proposed queries complement the functions of *gOpenMol* system. The queries help the user to analyze molecular structures and their physical and chemical properties. The users are interested in the interaction between certain atoms in the molecular structure and the queries allow them to be able to focus their attention on the motion of certain atoms. We view the result of the queries as the trajectories of molecules using *gOpenMol* software.

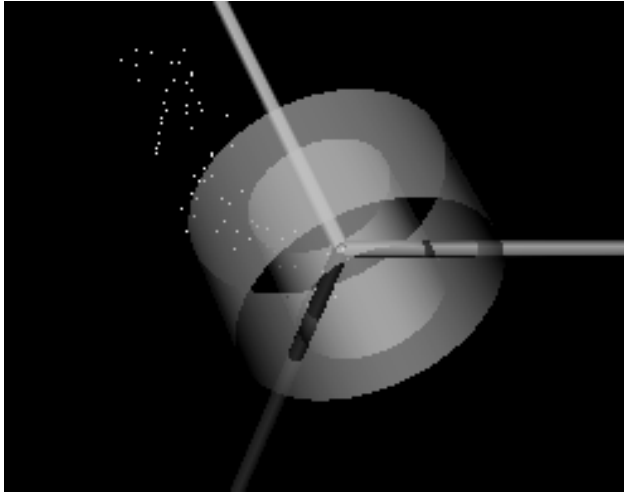


Figure 4. Result of the query

5. Conclusion and Future Work

In this paper, we introduced geometric approach to querying of time-dependent data in spatio-temporal, and temporal databases, and data warehouses. We extended the query model with time dimension to enable working with time-dependent data. We also implemented solid queries with time-dependent parameters and location that can find its further application in domain of molecular dynamics or other potential application areas where time-dependent data are studied.

The proposed query model can be also easily extended to query time-dependent solids. Now, we are working on time-dependent data representation model to allow implementation of efficient spatial access methods and structures.

We are planning to integrate our visual querying interface with the molecular dynamics system *gOpenMol*. Other plans include implementing the proposed geometric query model in VRML to be able to work in collaborative environment of Cyberworlds.

6. References

- [1] M. H. Dunham, *Data Mining Introductory and Advanced Topics*, Person Education, New Jersey, 2003.
- [2] D.A. Keim, "Information Visualization and Visual Data Mining", *IEEE Transactions on Visualization and Computer Graphics*, 8(1), 2002, pp. 1-8.
- [3] R. H. Güting et al, "A Foundation for Representing and Querying Moving Objects", *ACM Transaction on Database Systems*, Vol. 25, No. 1, March 2000, pp. 1-42.

[4] M. Erwig, M. Schneider, "Developments in Spatio-Temporal Query Languages", *IEEE Int. Workshop on Spatio-Temporal Data Models and Languages (STDML)*, 1999, pp. 441-449.

[5] O. Sourina., S.H Boey, "Geometric Query Types for Data Retrieval in Relational Databases", *Data & Knowledge Engineering, Elsevier Science B.V.*, 27(2), 1998, pp. 207-229.

[6] J. Bloomenthal, *An Introduction to Implicit Surfaces*, Morgan-Kaufmann, San Francisco, CA, 1997.

[7] O. Sourina, and L. Dongquan, "Geometric approach to clustering and querying in databases and warehouses", In *Proc. of Cyberworlds 2003*, Singapore, Dec. 2003, pp. 326-333.

[8] A. Ricci A., "A constructive geometry for computer graphics", *The Computer Journal*, Vol. 16, 2, 1973, pp. 157-160.

[9] W. Schroeder, K. Martin, B. Loresen, *The Visualization Toolkit*, Prentice Hall, 1998.