Spatiotemporal Reasoning about Epidemiological Data

Peter Revesz [*], Shasha Wu [**]

Keywords

Epidemiology, knowledge-base, recursive definition, spatiotemporal data, visualization, West Nile Virus.

Abstract

Objective. In this article, we propose new methods to visualize and reason about spatiotemporal epidemiological data.

Background. Efficient computerized reasoning about epidemics is important to public health and national security, but it is a difficult task because epidemiological data are usually spatiotemporal, recursive, and fast changing hence hard to handle in traditional relational databases and geographic information systems.

Methodology. We describe the general methods of how to (1) store epidemiological data in constraint databases, (2) handle recursive epidemiological definitions, and (3) efficiently reason about epidemiological data based on recursive and non-recursive SQL queries.

Results. We implement a particular epidemiological system called WeNiVIS that enables the visual tracking of and reasoning about the spread of the West Nile Virus epidemic in Pennsylvania. In the system, users can do many interesting reasonings based on the spatiotemporal dataset and the recursively defined risk evaluation function through the SQL query interfaces.

Conclusions. In this article, the WeNiVIS system is used to visualize and reason about the spread of West Nile Virus in Pennsylvania as a sample application. Beside this particular case, the general methodology used in the implementation of the system is also appropriate for many other applications. Our general solution for reasoning about epidemics and related spatiotemporal phenomena enables one to solve many problems similar to WNV without much modification.

1 INTRODUCTION

Infectious disease outbreaks are critical threats to public health and national security [5]. With greatly expanded travel and trade, infectious diseases can quickly spread across large areas causing major epidemics.

Efficient computerized reasoning about epidemics is essential to detect their outbreak and nature, to provide fast medical aid to affected people and animals, to prevent their further spread, and to manage them in other ways.

Several characteristics of epidemics make them special in terms of computer reasoning needs. First, epidemiological data are usually some kind of spatiotemporal data, that is, they have a spatial distribution that changes over time. Second, epidemiological data are recursive in nature. This means that the best predictions of the spread of infections are based on earlier situations. Third, we need a fast response from any knowledge-base that contains epidemiological data. A flexible information system that can be easily modified to model new epidemics is critical in assisting people to handle the outbreaks of new diseases.

The above three characteristics in combination pose a difficult problem. Geographic information systems generally can represent only static objects that do not change over time, or if they change, then they change only slowly, for example, the population density of counties. Such a slow change may be represented in a geographic information system by a limited number of separate maps. However, continuous change over time is not easy to represent and is hard to reason about in geographic information systems.

We propose new methods to visualize and reason about epidemiological data. The major contributions and novel features of our article are the following:

- **General method for recursively defined spatiotemporal models:** We propose a new general method to model a class of recursively defined spatiotemporal concepts, which appear in many research areas including epidemiology. In this article, we extend the definition in [24] to allow linear combination of the measurements of the indicators and a different time delay for each indicator.
- **Recursive epidemiological definitions:**

* Corresponding author. Complete Address: Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588, USA. Tel.: +1-402-472-3488; fax:+1-402-472-7767. email: revesz@cse.unl.edu

**Complete Address: Department of Computer Science, Spring Arbor University, Spring Arbor, MI 49283, USA. email: swu@arbor.edu. The work presented here was done while the author was at UNL.

We apply this new method to express the recursive epidemiological definitions and predictions about the spread of infectious diseases.

- **Implementation using recursive SQL:**
  The Prolog language is the choice for recursive definitions in many knowledge-base systems. However, Prolog is not good for querying spatiotemporal data. It is also less well-known than the widely-used Structured Query Language (SQL), which is the standard query language for both relational and constraint databases. The latest SQL standard added to the SQL language a form of recursion, enabling the expression of the needed recursive definitions. It is expected that the latest SQL standard will be implemented in all major relational database products. As part of our contributions, we also implemented for the first time in the MLPQ [23] constraint database system, which is one of the most sophisticated constraint database systems, the SQL recursive queries.

- **Epidemiological data stored in constraint databases:**
  Relational databases and geographic information systems can not easily manage epidemiological data because of their inherently spatiotemporal nature. Constraint databases [10, 12, 22], which are very suitable for spatiotemporal data, were proposed as extensions of both relational databases and geographic information systems. There are software tools that can export any relational database or geographic information system data into a constraint database [3, 2].

- **WeNiVIS: The West Nile Virus Information System:**
  We developed an example epidemic information system for reasoning about West Nile Virus infections. This system can show visually the spread of the epidemic and any other spatiotemporal data that may be generated by the system. We chose this example, because it has a typical infection pattern, it is currently still spreading through the United States, and data for it was readily available from Pennsylvania's West Nile Virus Control Program [19].

The rest of the article is organized as follows: Section 2 describes some basic concepts and related work. Section 3 describes the new general method for modeling recursively defined spatiotemporal concepts. Section 3.1 proposes a general recursive definition for spatiotemporal concepts. Section 3.2 describes the solution and optimization for the recursive definition using recursive SQL query language. Section 4 describes the source data we use for the West Nile Virus analysis (in Section 4.1), their interpolation and storage in a constraint database (in Section 4.2), and the West Nile Virus Information System (WeNiVIS) we developed for the WNV analysis (in Section 4.3). Section 5 presents major results and benefits of this project. Section 6 discusses some specific issues about our method and system. Finally, Section 7 gives some conclusions and directions for future work.

2 BASIC CONCEPTS AND RELATED WORK

2.1 Recursive Queries

We give only a brief introduction to recursive queries in relational databases [4, 6, 21, 26]. Figure 1 shows a relational database table that describes child-parent relationships. A recursive query on this table would be to find all the ancestors of *David*.

Family

Child	Parent
David	Andrew
David	Jane
Andrew	Scott
Andrew	Mary
Mary	Tracy
⋮	⋮

Fig. 1. Relationship of a family.

The latest ANSI (American National Standards Institute) SQL Language allows a form of recursion, enabling the expression of the above recursive query. We implement the recursive SQL for the first time in the MLPQ constraint database system. The syntax of the recursive SQL in the MLPQ system follows the latest SQL standard with only a minor modification.

A *non-recursive* SQL view definition is a statement of the form:

create view V_i **as** B_i;

where V_i is a view name with attributes and B_i is an SQL statement that uses only input relations (tables). Such B_is are called *basic SQL expressions*. A *recursive* SQL view definition has the form:

create view V_i **with recursive as** B_i **union** R_i;

where V_i is a view name with attributes. Here V_i is defined using the union of a basic SQL expression B_i and a *recursive SQL expression* R_i, which may contain a reference to V_i or other non-recursive and recursive views.

A sample recursive SQL query that finds all ancestors of *David* based on the table of Figure 1 can be expressed as follows:

Query 2.1 Find all ancestors of *David*:

```
create view DavidAncestors(Ancestor)  with recursive as
(select     Parent
from        Family
where       Child = "David")
union
(select     F.Parent
from        Family as F,  DavidAncestors as D
where       F.Child = D.Ancestor)
```

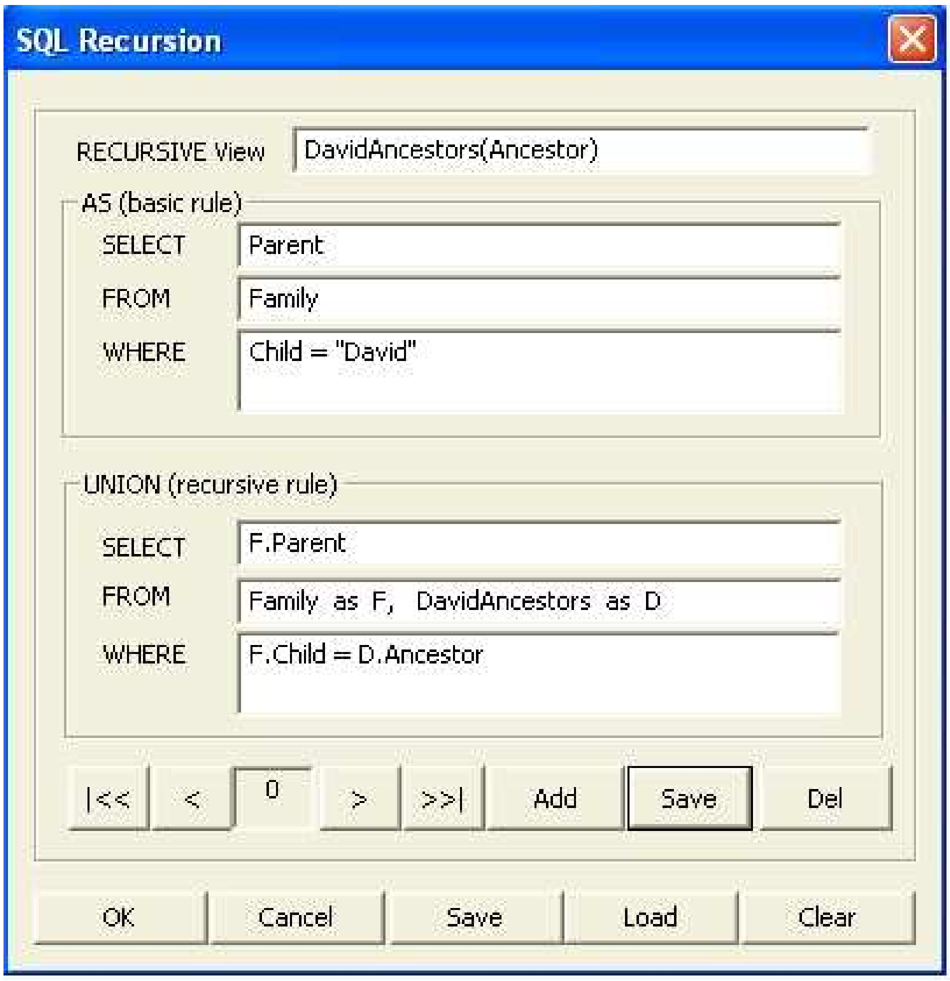
Figure 2 displays the implementation of Query 2.1 in the MLPQ constraint database system.

2.2 Constraint Databases Concepts

A *constraint database* is a finite set of constraint relations. A constraint relation is a finite set of *constraint tuples*, where each constraint tuple is a conjunction of *atomic constraints* using the same set of attribute variables [22]. Hence, constraints are hidden inside the constraint tables, and the users only need to understand the logical meaning of the constraint tables as an infinite set of constant tuples represented by the finite set of constraint tuples. Typical atomic constraints include linear or polynomial arithmetic constraints.

The MLPQ system is a constraint database system that implements rational linear constraint databases and queries. MLPQ is the abbreviation for *Management of Linear Programming Queries*. Among other functionalities, it supports both SQL and Datalog queries, and minimum/maximum aggregation operators over linear objective functions [23]. It is a suitable tool for representing, querying, and managing spatiotemporal constraint databases. Other constraint database systems include CCUBE [1], CQA/CDB [7], and DEDALE [9], which could be also used.

Li and Revesz [15] considered constraint-based visualization for spatiotemporal data but did not consider recursively defined concepts. Revesz and Wu [24] considered constraint-based visualization for recursively defined spatiotempo-


SQL Recursion

RECURSIVE View: DavidAncestors(Ancestor)

AS (basic rule)

SELECT: Parent

FROM: Family

WHERE: Child = "David"

UNION (recursive rule)

SELECT: F.Parent

FROM: Family as F, DavidAncestors as D

WHERE: F.Child = D.Ancestor

Fig. 2. User interface for recursive SQL in the MLPQ system.

ral data, but they only consider one indicator with a fixed time delay. That is too simple for real epidemiological problems and need to be extended. In epidemiology, one infectious disease commonly has several indicators (i.e. measurable disease carriers) and different indicators may have different effectiveness with different delay times.

2.3 Interpolation Methods

In a 2-D spatial problem, a point-based spatiotemporal relation has the schema of (x, y, t, w_1, w_2, $\ldots$, w_m), where the attributes (x, y) specify point locations, t specifies a time instance, and w_i $(1 \leq i \leq m)$ records the features of each location.

A point-based spatiotemporal data set only stores information of some sample points. To represent the features beyond those finite sample points, it is necessary to do spatiotemporal interpolation on them. A shape function based spatiotemporal method [14, 15] was used to interpolates and translates the original point-based spatiotemporal information into a *constraint relation.* Li and Revesz [16, 15] did an extensive comparison and proved shape functions to be the best over the Inverse Distance Weighting (IDW) [25] and Kriging [11, 17] interpolaters in a test example concerning house price estimation.

Figure 3 shows a point-based spatiotemporal data set consisting of the vertices shown there, and its "Delaunay Triangulation" network [8].

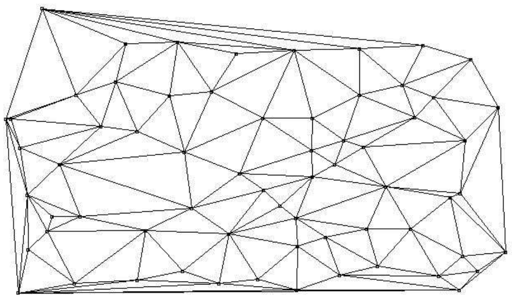


Fig. 3. The triangulated network of sample points in the state of Pennsylvania.

2.4 GIS Enhancement for Spatiotemporal Information

Geographic Information Systems (GIS) are designed for static data and need to be enhanced to be able to reason about spatiotemporal information [13, 29].

One such GIS enhancement is given by Theophilides et al. [28], who developed DYCAST, which is an epidemic spread prediction system based on spatiotemporal interpolation. The DYCAST system was used to predict human West Nile Virus infections based on dead bird surveillance data. However, the DYCAST system does not provide a flexible reasoning method.

Another GIS enhancement is given by Raffaetà et al. [20], who use MuTACLP, which is a temporal annotated constraint logic programming language. While in theory MuTACLP can describe spatial data by using constraints similarly to constraint databases [10, 12, 22], Raffaetà et al. [20] are only interested in using MuTACLP on top of a GIS. The temporal annotations are simple, that is, they allow only to declare some atomic formula is true at a certain time, true throughout a time interval, or true sometime during a time interval. MuTACLP is implemented based on Sicstus Prolog 3.8.3.

In contrast to MuTACLP, we use more complex temporal conditions, i.e., we allow any linear constraint on the spatial variables x and y and temporal variable t, and our implementation is based directly on the MLPQ [23] constraint database system.

3 METHODOLOGY

3.1 General Definition for Recursively Defined Spatiotemporal Concepts

Revesz and Wu proposed a general definition for recursively defined spatiotemporal concepts in [24]. Unfortunately, that definition is too limited for our current need, because it only deals with one indicator with fixed one unit time delay. In epidemiology, one infectious disease commonly has several indicators (i.e. measurable disease carriers) and different indicators may have different effectiveness. The animal indicators also may predict ahead of the human infection with different delay times. To consider these extra complications, we extend their definition as follows:

Definition 3.1 Let $M_i(x, y, t)$ represent the amount of indicator i measured at location (x, y) at time unit t. For each indicator i, let w_i be the effectiveness weight and d_i be the time delay to indicate property P. Then location (x, y) has property P during time unit t if

(1) $\sum w_i M_i(x, y, t - d_i) \ \geq \ k$ or
(2) $k_1 \ \leq \ \sum w_i M_i(x, y, t - d_i) < k$
and the location has property P during time unit $t - 1$.

Part (1) of Definition 3.1 says that property P holds at time t if the linear combination of measurements of the indicators at the appropriate previous times (i.e., with their respective time delays) is greater than some threshold value k. Part (2) says that P also holds in those areas where the same linear combination is only between k_1 and k but already had property P at time $t - 1$.

Example 3.1 The West Nile virus has four major types of disease indicators: wild bird as indicator 1, mosquito as indicator 2, chicken as indicator 3 and horse as indicator 4. Figure 4 suggests that the onset of human infections generally occurs three weeks later than the onset of wild bird infections, one week later than the onset of mosquito infections, about six weeks after the onset of chicken infections and almost at the same time as the horse infections. Hence, we can assign the time delay for these four indicators as follows:

$d_1 = 3, \ \ d_2 = 1, \ \ d_3 = 6, \ \ d_4 = 0$

Considering that big animals usually contain more virus than small animals contain, we may assign the effectiveness weight of WNV infection to the four major carriers according to their relative body sizes as follows:

$w_1 = 1, \ \ w_2 = 0.2, \ \ w_3 = 1.5, \ \ w_4 = 5$

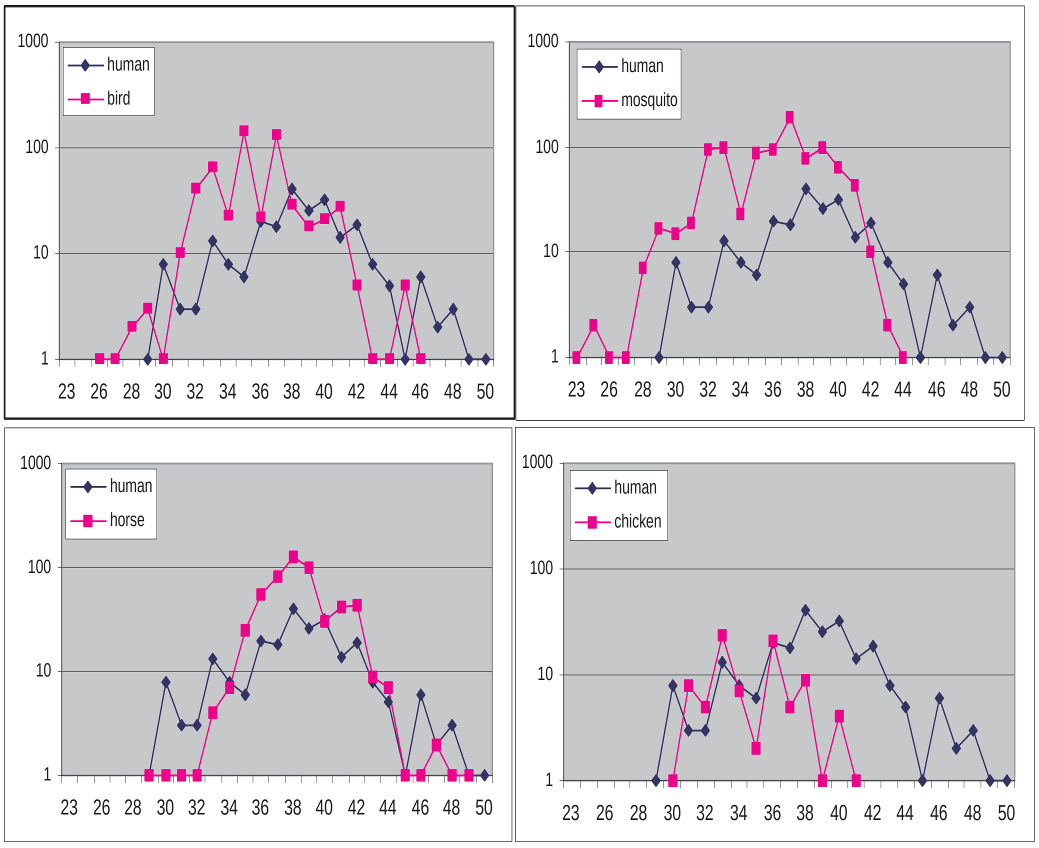


Fig. 4. The comparison of time lags between the infections on human and various types of animal hosts (*X-coordination represents the week in year 2002 and Y-coordination is the number of reported infectious cases).*

We assume that the infected animals reported at time $t - d_i$ are representative of the entire animal population at the same time and part of the unreported infected animals at that time may continue to be infected at least until time t.

Suppose we would like to find the areas on a map that have a high risk of human WNV infections at time t. Let $k = 8$ and $k_1 = 4$, and $M_i(x, y, t)$ be as in Definition 3.1.

First, we compute the linear combination of the measurements of the indicators for each area as follows:

$$\begin{aligned} w &= \sum w_i M_i(x, y, T - d_i) \\ &= M_1(x, y, T - 3) + 0.2 M_2(x, y, T - 1) + 1.5 M_3(x, y, T - 6) + 5 M_4(x, y, T) \end{aligned}$$

Then the area is at high risk of human WNV infections at week t if during week t it has

(1) $w \geq 8$ or
(2) $4 \leq w < 8$ *and* it is at high-risk during week $t-1$.

3.2 Solution and Optimization

The general solution for the problem defined in Definition 3.1 can be formally expressed as follows.

Given relations $M_i(x, y, t, m)$ where the value m represents the measurement of indicator i at location (x, y) at time t for each $1 \leq i \leq n$, let us define the following:

$$A = \{(x,y,t) \mid M_1(x,y,t-d_1,m_1) \wedge \ \ldots \ \wedge M_n(x,y,t-d_n,m_n) \wedge \\ w_1m_1 + \ldots + w_nm_n \geq k \ \}$$

$$B = \{(x,y,t) \mid M_1(x,y,t-d_1,m_1) \wedge \ \ldots \ \wedge M_n(x,y,t-d_n,m_n) \wedge \\ k_1 \leq w_1m_1 + \ldots + w_nm_n < k \ \}$$

Where A is the part of $M_1, \ldots, M_n$ where the linear combination of measurements of all indicators is greater or equal to k, and B is the part that the linear combination of measurements of all indicators is between k and k_1. The above definition can be implemented in the SQL query language as follows:

Query 3.1 SQL query for linear combination and time delay:

```
create view A(x, y, t) as
select      M_1.x,  M_1.y,  t
from        M_1, ..., M_n
where       w_1 M_1.m + ... + w_n M_n.m >= k,
            M_1.t = t - d_1,  ...,  M_n.t = t - d_n,
            M_1.x = ... = M_n.x,
            M_1.y = ... = M_n.y
```

Relation A returns the spatiotemporal locations (x, y, t) that satisfy part (1) of Definition 3.1.

```
create view  $B(x,y,t)$  as
select       $M_1.x,\ \ M_1.y,\ \ t$
from         $M_1, \ldots, M_n$
where        $k_1 \le w_1 M_1.m + \ldots + w_n M_n.m < k,$
             $M_1.t = t - d_1,\ \ \ldots,\ \ M_n.t = t - d_n,$
             $M_1.x = \ldots = M_n.x,$
             $M_1.y = \ldots = M_n.y$
```

Relation B returns the spatiotemporal locations (x,y,t) that satisfy the first condition of part (2) of Definition 3.1.

Based on relations A and B and Definition 3.1, we can define the areas having property P at time t as follows:

$$P = \{(x,y,t) \ \mid\ A(x,y,t) \ \ \vee (B(x,y,t) \wedge P(x,y,t-1))\}$$

We found that a direct implementation of the above is very inefficient. To reduce the computational complexity, we express P by a logically equivalent formula that is easier to evaluate [24]:

Theorem 3.1

$$P = \{(x,y,t) \ \mid A(x,y,t) \ \vee \ \left(\bigvee_{m=1}^{+\infty} (C(x,y,t,m-1) \ \wedge \ A(x,y,t-m)) \right) \}$$

where

$$C = \{(x,y,t,m) \mid (m=0 \ \wedge \ B(x,y,t)) \ \vee \\ (m \ge 1 \ \wedge \ B(x,y,t-m) \ \wedge \ C(x,y,t,m-1))\}$$

Proof: First, we can prove for any $m \ge 1$ that

$$C(x,y,t,m) = \bigwedge_{i=0}^{m} B(x,y,t-i) \tag{1}$$

as follows:

$$\begin{aligned}
C(x,y,t,m) &= B(x,y,t-m) \ \wedge \ C(x,y,t,m-1) \\
&= B(x,y,t-m) \ \wedge \ B(x,y,t-m+1) \ \wedge \ C(x,y,t,m-2) \\
&= B(x,y,t-m) \ \wedge \ \ldots \wedge \ B(x,y,t-1) \ \wedge \ C(x,y,t,0) \\
&= B(x,y,t-m) \ \wedge \ \ldots \wedge \ B(x,y,t-1) \ \wedge \ B(x,y,t) \\
&= \textstyle\bigwedge_{i=0}^{m} B(x,y,t-i)
\end{aligned}$$

Second, by expanding $P(x,y,t-1)$ in the definition of P we get:

$$\begin{aligned}
P = \{(x,y,t) \ | \ & A(x,y,t) \ \vee \ [B(x,y,t) \ \wedge \\
& (A(x,y,t-1) \vee (B(x,y,t-1) \wedge P(x,y,t-2)))]\} \\
= \{(x,y,t) \ | \ & A(x,y,t) \ \vee \ [B(x,y,t) \wedge A(x,y,t-1)] \ \vee \\
& [B(x,y,t) \wedge B(x,y,t-1) \wedge P(x,y,t-2)]\}
\end{aligned}$$

We can continue to expand $P(x,y,t-2)$ and simplify it as follows:

$$\begin{aligned}
P = \{(x,y,t) \ | \ & A(x,y,t) \ \vee \\
& \left(\left(\bigwedge_{i=0}^{0} B(x,y,t-i)\right) \wedge A(x,y,t-1)\right) \vee \\
& \ldots \vee \left(\left(\bigwedge_{i=0}^{m-1} B(x,y,t-i)\right) \wedge P(x,y,t-m)\right)\}
\end{aligned}$$

Using Equation (1), the above can be further simplified as:

$$\begin{aligned}
P = \{(x,y,t) \ | \ & A(x,y,t) \ \vee \\
& (C(x,y,t,0) \wedge A(x,y,t-1)) \vee \\
& \ldots \vee (C(x,y,t,m-1) \wedge P(x,y,t-m))\}
\end{aligned}$$

Finally, the right hand side of the formula can be expanded to:

$$A(x,y,t) \ \vee \ \left(\bigvee_{m=0}^{+\infty} (C(x,y,t,m) \ \wedge \ A(x,y,t-m-1))\right) \qquad \square$$

Based on Theorem (3.1), we can express the optimized recursive SQL queries as follows:

Query 3.2 The recursive SQL query expressing relation C in Theorem (3.1) is:

```
create view  C(x, y, t, m)  with  recursive  as
(select      x, y, t, m
from         B
where        m = 0)
union
(select      x, y, t, m
from         B, C
where        m ≥ 1, m ≤ M,
             B.x = x, B.y = y, B.t = t − m,
             C.x = x, C.y = y, C.t = t, C.m = m − 1)
```

where relations $A(x, y, t)$ and $B(x, y, t)$ are defined in Query 3.1. Assume $C_k = \{(x, y, t) \mid C(x, y, t, k)\}$, then we have $C_j \subseteq C_i$ for all $1 \leq i < j$. That means for each fixed time t the area of $C(x, y, t, m)$ monotonously decreases as m increases.

To set the boundary of m, we introduce $M \geq 1$ that is used to ensure the termination of the recursive evaluation process. The bigger M is, the more accurate the result is, but the more calculation is required. Next, the recursive SQL query corresponding to relation P is:

```
create view  P(x, y, t)  as
(select      x, y, t
from         A)
union
(select      C.x, C.y, C.t
from         A, C
where        C.x = A.x, C.y = A.y, A.t = C.t − C.m − 1)
```

4 THE WENIVIS SYSTEM

4.1 The West Nile Virus Data

West Nile Virus (WNV) was originally discovered in the West Nile district of Uganda in 1937. It causes infection and fevers in humans in Africa, West Asia, and the Middle East. The first report of WNV in the United States was found in 1999 in New York City [18]. Since then, the disease has spread across the United States. In 2003, WNV activity occurred in 46 states and caused illness in over 9,800 people [27].

WNV is transmitted to humans through mosquito bites. Mosquitoes become infected when they feed on infected birds that have high levels of WNV in their blood. Infected mosquitoes can then transmit WNV when they feed on humans or other animals [27].

We obtained data on the spread of WNV in Pennsylvania in 2003 from Pennsylvania's West Nile Virus Control Program [19]. The data include dead wild birds, mosquitoes, sentinel chickens, equine (horse) veterinary and confirmed human cases of WNV as explained below.

- **Dead wild bird:** In Pennsylvania's WNV Control Program, the dead birds were collected by passive surveillance, relying on public reporting through telephone and Internet. Dead birds and infected birds that display erratic behavior are highly visible for casual observers to identify and report in the areas where WNV may be active. When a dead bird is sighted, the information about that bird and its location is recorded. Then samples from each bird are tested for WNV and the results are recorded in the database.
- **Mosquito:** Mosquitoes were routinely collected in surveillance locations. All mosquitoes collected in one effort create a sample of mosquitoes. Only adult mosquitoes in the sample are tested for WNV.
- **Sentinel chicken:** Pennsylvania's surveillance system includes sentinel chickens. Several flocks are housed near the areas that have dense human populations and stagnant water sources. Those flocks are used by the medical experts to monitor the presence of the virus. Samples from these sentinels are collected weekly and tested for WNV.
- **Equine:** Equine diagnostic blood samples submitted by veterinarians across the state are also tested for WNV.

Since the datasets are not recorded during the weekends, the data are summarized to weekly data to generate a continuous surveillance dataset.

We first compare the onset of several kinds of animal infection hosts with the onset of human infections. To show the time relationship between the number of various animal disease carriers found and the number of human infection cases reported, we show in four separate charts in Figure 4 the number of the four animal diseases in red (gray) curves and the number of human cases in blue (black) curves. Figure 4 shows that there is a time lag of about six weeks between the onset of sentinel chicken infections and human cases. Similarly, there is a time lag of about three, one, and zero weeks between the infected wild bird, mosquito, and equine veterinary cases with respect to the human cases. Hence while Figure 4 shows that each of the three types of animal WNV infections are strongly related to the human WNV infections, the various animal cases provide different advance warnings of human WNV epidemic outbreaks.

4.2 Epidemiological Data in Constraint Databases

A point-based spatiotemporal data set only stores information of some sample points at some sample times. That is usually what one can obtain as the raw data for infectious diseases. To represent the features beyond those finite spatiotemporal points, it is necessary to do some *spatiotemporal interpolation* on them. Interpolation requires some basic assumptions about the nature of the point data set. Theophilides et al. [28] makes the following interpolation assumptions:

(1) West Nile Virus is a continuous phenomenon across space;
(2) Humans are infected at their resident places;
(3) Nonrandom space-time interaction of bird deaths is attributed to West Nile Virus infection;
(4) Each dead bird has an equal opportunity of being reported;

We make similar assumptions. We differ from [28] by applying instead of Knox spatiotemporal interpolation a 2-D shape function-based interpolation method, which Li and Revesz [15] found to be the most reliable among several well-known spatial and spatiotemporal interpolation methods. It is easily implementable in constraint database systems, as we will see later.

Before the interpolation, we need to get some data point from each county in Pennsylvania. We have the positions of 102 cities and towns of Pennsylvania. We pick as a sample point the biggest city of each county based on the 1990 city population census in Pennsylvania. If a county is too small to have any cities on the list, we arbitrarily pick the center of the county as the sample point.

Figure 5 shows a part of the constraint relation that describes the result of a linear shape function-based interpolation [15]. The constraint relation contains many constraint tuples (rows).

Each constraint tuple contains three or four constraints. The first three constraints represent the area of a triangle as the intersection of three linear inequality constraints over x and y. The fourth constraint is a linear equation that represents the danger level n of WNV infection to humans. The danger level is calculated by a spatiotemporal interpolation over all locations (x, y) and times t of the known infection data. We can predict the number of human WNV infections by some product of the danger level and the total human population in that location.

In each tuple the *week* attribute always represents the time measured in weeks past January 1st, 2001, hence the week starting January 1st, 2001 would be week 1, the week starting January 8th, 2001 would be week 2 etc. Finally, we also give a unique *id* value to each triangular area for easy identification.

WNV Birds

Id	Lat	Long	Week	#WNVbirds	
1	x	y	122	m	$x - 0.14y \leq -85.83$, $x + 0.4y \leq -64.13$, $-x - 0.1y \leq 76.54$, $m = 0$
2	x	y	122	m	$x + 1314.06y \leq 52284.18$, $-x + 126.09y \leq 5101.43$, $x - 211.55y \leq -8504.74$, $m = 0$
⋮	⋮	⋮	⋮	⋮	⋮
65	x	y	130	m	$-x + 1.73y \leq 147.76$, $-x - 4.09y \leq -85.06$, $x - 0.26y \leq -88.39$, $m + 2.28x - 0.59y = -201.29$
⋮	⋮	⋮	⋮	⋮	⋮

Fig. 5. Weekly number of WNV-infected birds in Pennsylvania during 2003 represented in a constraint relation.

4.3 Implementation of the WeNiVIS System

The West Nile Virus Information System (WeNiVIS) is an epidemiological information system designed to manage the spatiotemporal WNV information. It is a two-layer client/server system built on top of the MLPQ constraint database system and has many unique functionalities.

The WeNiVIS system has five major components shown in Figure 6 as follows:

(1) **Recursive reasoning interface** provides a convenient user interface to define the recursive epidemiology concepts.
(2) **Visualization window** can accept and display the spatial result of the reasoning in a set of snapshots or animation.
(3) **Time navigating bar** can be used to track visually the spread of WNV over time. It is especially helpful when users have to track and compare several maps with different indicators using different time lags.
(4) **SQL query interfaces** are helpful to guide users in generating correct SQL queries.
(5) **Socket-based communication module** takes care of the data communication between the WeNiVIS system and the MLPQ system.

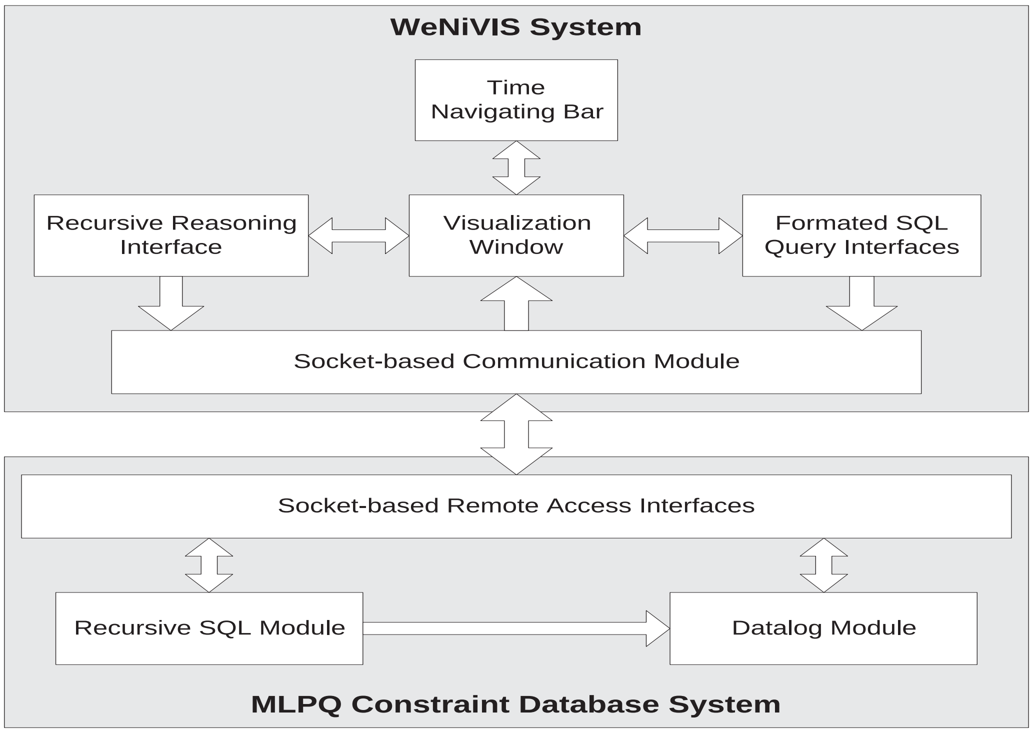
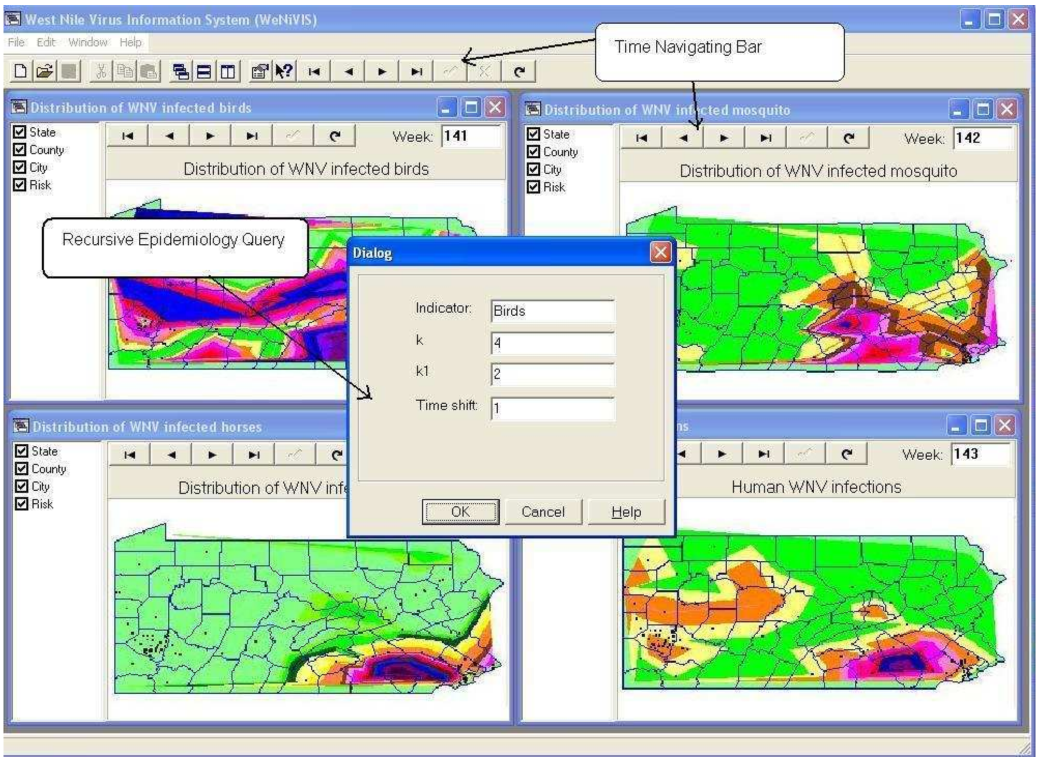

WeNiVIS System
Time Navigating Bar
Recursive Reasoning Interface
Visualization Window
Formated SQL Query Interfaces
Socket-based Communication Module
Socket-based Remote Access Interfaces
Recursive SQL Module
Datalog Module
MLPQ Constraint Database System

Fig. 6. The Software architecture of the WeNiVIS system.

Figure 7 shows the interface of the WeNiVIS system. The user can generate many visualization windows in the frame window. In Figure 7, each visualization window represents one indicator of the WNV infections.


West Nile Virus Information System (WeNiVIS)
File Edit Window Help
Time Navigating Bar
Distribution of WNV infected birds
Week: 141
Distribution of WNV infected mosquito
Week: 142
Recursive Epidemiology Query
Distribution of WNV infected horses
Human WNV infections
Week: 143
State
County
City
Risk
Dialog
OK
Cancel
Help

Indicator: Birds
k: 4
k1: 2
Time shift: 1

Fig. 7. The WeNiVIS system in analyzing WNV infections.

As shown in the center of Figure 7, the recursive reasoning dialog box provides a simple interface for the user to generate complex recursive queries. It only asks for the values of several critical parameters and can automatically generate a recursive query based on the general format defined in Query 3.2. The new recursive query is sent to and evaluated by the MLPQ system.

The WeNiVIS system has two kinds of time navigating bars. The local time navigating bar on top of each visualization window is used to choose the time of the map displayed in that window. The changing of the time in each visualization window by its own time navigating bar is independent of the time in the other windows. On the other hand, the global time navigating bar in the frame window can be used to change the time of all visualization windows within that frame window at the same speed. For example, the user can first set the time of one visualization window as 122 and the time of another window as 125 by their own time navigating bar. Then the user can use the global time navigating bar to browse both of these two windows simultaneously. This function is helpful when the user wants to check visually the effects of the distribution of the several indicators, which may have different time delays with respect to the target property.

The SQL query interface has a fixed format for different kind of SQL queries. The WeNiVIS system supports basic, aggregation, set, nested and recursive queries on constraint databases.

5 RESULTS

5.1 Flexible User Interfaces with a High-Level Language

In the WeNiVIS system, users can do many interesting reasonings based on the spatiotemporal dataset and the recursively defined risk evaluation through the SQL query interfaces. We have five tables defined as follows:

- **Risk(x, y, t, w)** is the result of the recursive risk evaluation based on Definition 3.1 and Query 3.2. It stores the predicted risk value as w at location (x, y) during time unit t.
- **City(id, x, y, name, pop)** stores information about the cities in Pennsylvania. Attribute *pop* is the population of the city based on the census of year 1990.
- **Event(id, title, organizer, t, cid)** stores information about conferences and other events that are scheduled to be held in city *cid* at time t by the $organizer$.
- M_1**(x, y, t, m)** is the constraint-based interpolation result of the measurements of WNV-infected dead birds.
- M_2**(x, y, t, m)** is the constraint-based interpolation result of the measurements of WNV-infected mosquitoes.

Next we describe some sample queries that can be executed in the WeNiVIS system.

Query 5.1 Return the regions that have a high risk of human WNV infections at time t according to the measurements of both infected birds and mosquitoes.

```
create view  $A(x, y, t)$  as
select       $x,\ y,\ t$
from         $M_1,\ M_2$
where        $M_1.t = t - 3, \quad M_2.t = t - 1,$
             $M_1.x = x, \quad M_2.x = x, \quad M_1.y = y, \quad M_2.y = y,$
             $M_1.m \ + \ 0.2M_2.m \ \geq \ 1.4$

create view  $B(x, y, t)$  as
select       $x,\ y,\ t$
from         $M_1,\ M_2$
where        $M_1.t \ = \ t - 3, \quad M_2.t \ = \ t - 1,$
             $M_1.x = x, \quad M_2.x = x, \quad M_1.y = y, \quad M_2.y = y,$
             $0.4 \ \leq \ M_1.m \ + \ 0.2M_2.m \ < \ 1.4$
```

In the above queries, we set $k = 1.4$, $k_1 = 0.4$, and $M = 4$ as the upper bound of variable m and directly execute the optimized recursive SQL Query 3.2 to compute relations C and P. Of course, epidemiology experts may choose any other values as desired. Figure 8 is the output of this query at time $t = 139$ and the actual human cases at that time.

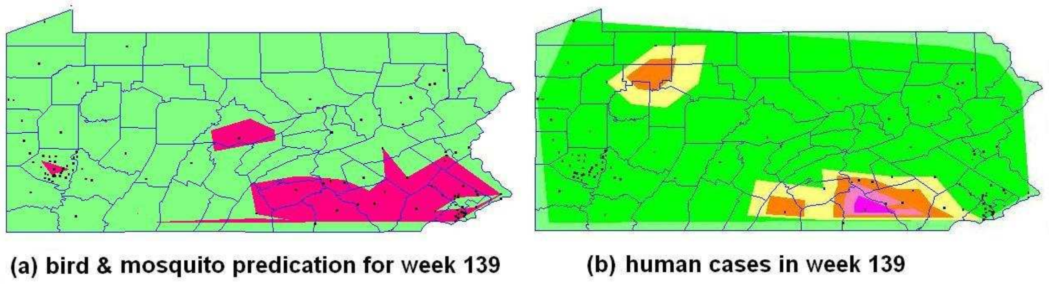

(a) bird & mosquito predication for week 139
(b) human cases in week 139

Fig. 8. The prediction of human West Nile Virus infections in week 139 based on bird and mosquito indicators.

Query 5.2 Pennsylvania health officials want to find and warn the organizers of the events that are scheduled to be held in a city when it is at a high risk of WNV infection.

```
create view  $WarnList(organizer)$  as
select       $E.organizer$
from         $City$ as $C$,  $Event$ as $E$,  $Risk$ as $R$
where        $C.id = E.cid,\ C.x = R.x,\ C.y = R.y,\ R.t = E.t,\ R.w \geq 1$
```

Query 5.3 Find the total population of the major cities in the areas that is at high risk of WNV infection in week 141.

```
create view  $InDanger(population)$  as
select       sum$(C.pop)$
from         $City$ as $C$,  $Risk$ as $R$
where        $C.x = R.x,\ C.y = R.y,\ R.t = 141,\ R.w \geq 1$
```

Query 5.4 A big sport event is scheduled to be held in Pennsylvania at week 141. Any city with more than 50,000 population in the state can host the event. People want to know which city will be safe from WNV infections during the event time.

```
create view CityList(id, name) as
select      id, name
from        City
where       pop >= 50000, id not in
            (select C.id
             from   City as C, Risk as R
             where  C.x = R.x, C.y = R.y, R.t = 141, R.w = 1)
```

The above queries are easy to understand and are efficiently evaluated by the WeNiVIS system. Comparing the size and complexity of a C++ or Java program needed to solve the same problem based on relational databases, SQL and constraint databases provide a more concise and manageable approach. A simple and independent query solution makes the program easy to understand and maintain.

5.2 Enhancement of Tracking and Reasoning about Epidemics

The WeNiVIS system is used in analyzing the spread of the West Nile Virus epidemic in Pennsylvania. Figure 9 displays the distribution of infected animals and humans for week 139 using a color band display with darker colors meaning more infections. We have four different small maps. The distribution of infected wild birds is shown in the upper-left, mosquitoes in the upper-right, horses in the lower-left, and humans in the lower-right window. The figure shows that the four cases are only weakly correlated with each other during the same week. Hence a time delay is needed in the analysis.

We already mentioned in Section 4.1 that the dead wild bird data are followed by the human cases after about three weeks. Figure 10 shows on the left side the predicted high-risk areas based on the infected wild bird data alone and on the right side the actual observed human cases. In generating the predictions we used Definition 3.1 with values $k_1 = 0.4$ and $k = 1.4$. We found these values by experiments. The smaller values of k_1 and k will generally yield more areas, but they may overpredict the risks and be less accurate.

In this article, the WeNiVIS system is used to visualize and reason about the spread of West Nile Virus in Pennsylvania as a sample application. Beside this particular case, the general methodology used in the implementation of the system is also appropriate for many other applications.

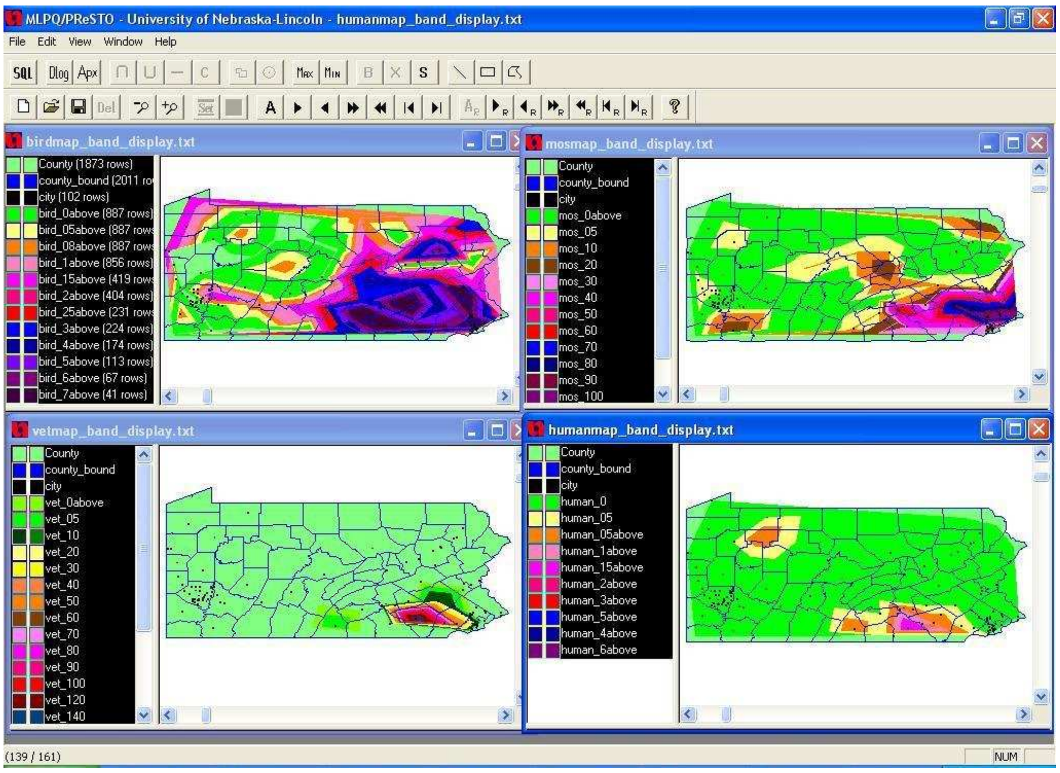


Fig. 9. The distribution of West Nile Virus infections in week 139 (upper-left: birds; upper-right: mosquitoes; lower-left: horses; lower-right: human).

First, with the additional data available, the system can be directly applied to any other states in which West Nile Virus may occur. We only need to translate the new state's data into a constraint database format, similarly as we have done for the Pennsylvania data set.

Second, based on our methodology, similar tracking and reasoning systems can be implemented for other epidemics besides the West Nile Virus epidemic.

Third, there are many other diseases that have recursive spread characteristics that are similar to those of epidemics. For example, some inherited diseases can be naturally defined by recursive definitions. The system can be also used to track and reason about many of those diseases too.

Finally, the recursive definition seems also natural for many other problems in medicine. For example, the current status of a chronic patient is closely related to his/her previous status. Our methods can be adapted to those situations too.

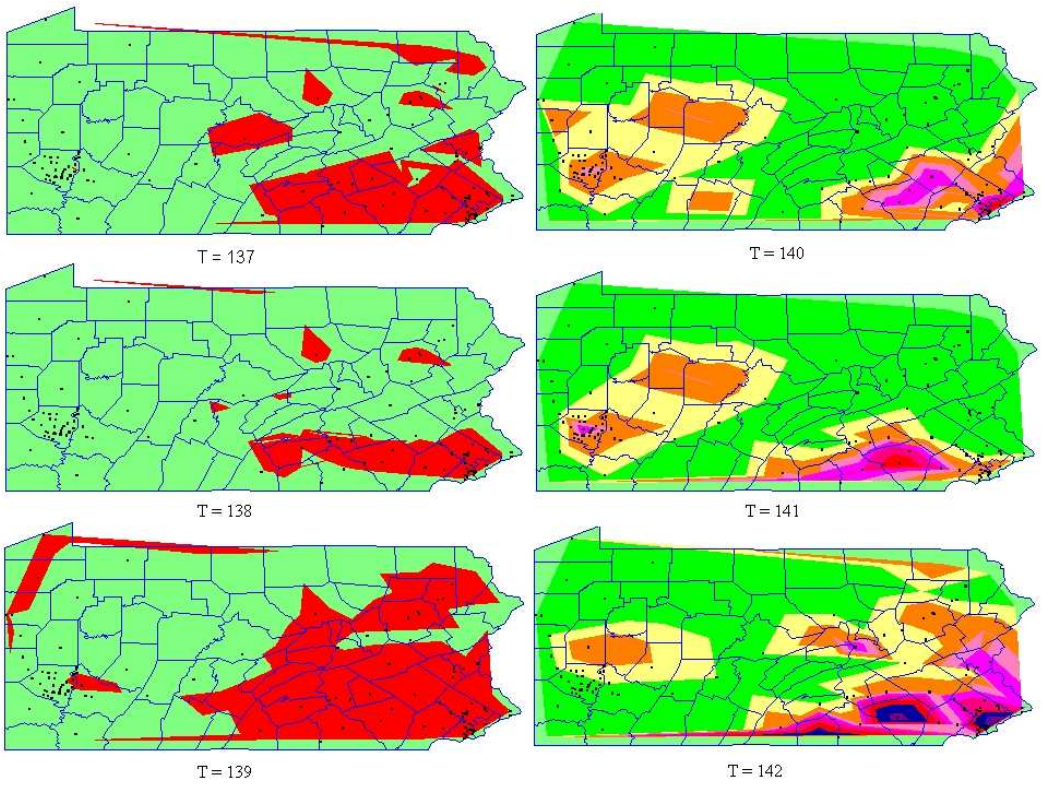


Fig. 10. Left side: The predicted high-risk areas (dark color areas) based on the infected wild bird data. Right side: The actual distribution of human infections (darker color areas mean more infections) three weeks after the wild bird data to its left.

6 DISCUSSION

Although the WeNiVIS system is implemented and tested on the WNV data of Pennsylvania, there are several general issues that need to be mentioned.

First, in the spatiotemporal interpolation process, we made four interpolation assumptions in Section 4.2 similar to Theophilides et al. [28]. However, we have to point out the sensitivity of these assumptions to the actual data being interpreted. Theophilides et al. [28] applied their method within New York City, where each district has a high density. Since New York City is heavily interconnected with a huge commuter population, assumption (2) may not hold.

We apply our interpolation over the whole state of Pennsylvania with a high variation of population density with the Philadelphia and the Pittsburgh areas having higher and the middle areas lower population densities. In this case, assumption (2) may be more reasonable, but assumption (4) may be less reasonable. Indeed, it is more reasonable to assume that in areas with a higher population density, a dead bird has a higher “opportunity” or chance of being reported than a dead bird in a low density area.

However, in this article our intent is not to make a medically valid interpolation but to illustrate some methodologies and leave it to the medical experts to test the methodology with empirical data and decide whether it is applicable to the particular epidemics that they study.

Second, we only consider the relative size of each kind of infected animal in assigning the effectiveness weight for it in Example 3.1. Alternatively, the effectiveness weight for a type of animal may be based on how close it generally lives to human beings. Intuitively, the closer an animal lives to human beings, the more likely it is to infect them.

Third, for spatiotemporal applications, recursive queries are not expressible using the basic query languages of GIS systems. Some relational databases and knowledge-base systems provide recursive queries, but they do not provide spatiotemporal data representation. Hence the visualization of recursively defined concepts cannot be easily handled by these systems. They would usually require some special functions to be written in a programming language like C or C++ and added to a library. In contrast, our system only uses standard SQL queries to solve the problem. Therefore, the program is a simple, declarative, and high-level query that is easy to maintain. This feature is important, because the requirement of visualizing recursively defined concepts on spatiotemporal data is frequent enough to need a general and simple solution method.

To manage a new epidemic in the WeNiVIS system, only the following modifications need to be made:

- Generate new base map in constraint databases: a software named *cdbdump* can be used to translate ArcGIS map file to a constraint database format.
- Interpolate measurements of indicators using shape-function based interpolation methods implemented in *cdbdump*.
- Define and examine the parameters k and k_1 in the SQL query to model the new epidemic.
- Apply the model to predict the new epidemic and check its correctness.

Based on our general definition for recursively defined spatiotemporal concepts and recursive SQL query solution, the WeNiVIS system can be easily extended to handle different epidemics.

7 CONCLUSION

Our general solution for reasoning about epidemics and related spatiotemporal phenomena enables one to solve many problems similar to WNV without much modification. The user does not need to search for different ad hoc solutions for each specific epidemiological case.

There are still many interesting directions for future work. For example, we are currently extending the WeNiVIS system by allowing the user to specify in a convenient way the color of each overlay object. We are also expanding the *recursive SQL generator* so that the user can select an object and calculate the growth in its area between two different time instances. Besides improving WeNiVIS, we are also branching out to other epidemic applications.

These and other issues remain interesting research topics for the future. We provide free copies of the WeNiVIS system to researchers and potential users who would like to try it out on WNV or other similar infectious disease data.

Acknowledgments

This research was supported in part by USA NSF grant EIA-0091530. The first author was also supported by an Alexander von Humboldt Research Fellowship.

References

[1] A. Brodsky, V. E. Segal, J. Chen, and P. A. Exarkhopoulo. The CCUBE constraint object-oriented database system. In Alex Delis, Christos Faloutsos, and Shahram Ghandeharizadeh, editors, *ACM SIGMOD International Conference on Management of Data*, pages 577–579, New York, 1999. ACM Press.

[2] J. Chomicki and P. Revesz. Constraint-based interoperability of spatiotemporal databases. *Geoinformatica*, 3(3):211–243, 1999.

[3] Jan Chomicki, Sofie Haesevoets, Bart Kuijpers, and Peter Revesz. Classes of spatio-temporal objects and their closure properties. *Annals of Mathematics and Artificial Intelligence*, 39(4):431–461, 2003.

[4] E. F. Codd. A relational model for large shared data banks. *Communications of the ACM*, 13(6):377–87, 1970.

[5] L. Damianos, J. Ponte, S. Wohlever, F. Reeder, D. Day, G. Wilson, and L. Hirschman. MiTAP for bio-security: a case study. *AI Magazine*, 23(4):13–29, 2002.

[6] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems - Fourth Edition.* Addison-Wesley, New York, 2003.

[7] D. Q. Goldin, A. Kutlu, M. Song, and F. Yang. The constraint database framework: lessons learned from CQA/CDB. In Umeshwar Dayal, Krithi Ramamritham, and T. M. Vijayaraman, editors, *International Conference on Data Engineering*, pages 735–737, Bangalore, India, 2003. IEEE Computer Society.

[8] J. E. Goodman and J. O'Rourke. *Handbook of Discrete and Computational Geometry.* CRC Press, Boca Raton, 1997.

[9] S. Grumbach, P. Rigaux, and L. Segoufin. The DEDALE system for complex spatial queries. In Laura M. Haas and Ashutosh Tiwary, editors, *ACM SIGMOD International Conference on Management of Data*, pages 213–24, New York, 1998. ACM Press.

[10] P. C. Kanellakis, G. M. Kuper, and P. Z. Revesz. Constraint query languages. *Journal of Computer and System Sciences*, 51(1):26–52, 1995.

[11] D. G. Krige. A statistical approach to some mine valuations and allied problems at the witwatersrand. Master's thesis, University of Witwatersrand, South Africa, 1951.

[12] G. M. Kuper, L. Libkin, and J. Paredaens, editors. *Constraint Databases.* Springer-Verlag, Heidelberg, 2000.

[13] G. Langran. *Time in Geographical Information Systems.* Taylor & Francis, London, 1992.

[14] L. Li, Y. Li, and R. Piltner. A new shape function based spatiotemporal interpolation method. In B. Kuijpers and P. Revesz, editors, *1st International Symposium on the Applications of Constraint Databases*, number 3074 in Lecture Notes in Computer Science, pages 25–40, Heidelberg, 2004. Springer.

[15] L. Li and P. Revesz. Interpolation methods for spatio-temporal geographic data. *Computers, Environment and Urban Systems*, 28(3):201–227, 2004.

[16] Lixin Li and Peter Revesz. A comparison of spatio-temporal interpolation methods. In Max J. Egenhofer and David M. Mark, editors, *Proceedings of the Second International Conference on Geographic Information Science*, volume 2478 of *Lecture Notes in Computer Science*, pages 145–160, Heidelberg, 2002. Springer-Verlag.

[17] G. Matheron. The theory of regionalized variables and its applications. *Les Cahiers du Centre de Morphologie Mathematique de Fontainebleau 5*, page 221, 1971.

[18] D. Nash, F. Mostashari, A. Fine, and et al. The outbreak of West Nile virus infection in the New York City area in 1999. *The New England Journal of Medicine*, 344(24):1807–1814, 2001.

[19] Pennsylvania's West Nile Virus Control Program. West Nile virus surveillance maps at http://www.westnile.state.pa.us/surv.htm, (Accessed: 10 December, 2004).

[20] A. Raffaetà, F. Turini, and C. Renso. Enhancing GISs for spatio-temporal

reasoning. In Agnès Voisard and Shu-Ching Chen, editors, *ACM-GIS'02: Proceedings of the 10th ACM international symposium on Advances in geographic information systems*, pages 42–48, New York, 2002. ACM Press.

[21] R. Ramakrishnan. *Database Management Systems.* McGraw-Hill, New York, 1998.

[22] P. Revesz. *Introduction to Constraint Databases.* Springer, New York, 2002.

[23] P. Revesz, R. Chen, P. Kanjamala, Y. Li, Y. Liu, and Y. Wang. The MLPQ/GIS constraint database system. In Weidong Chen, Jeffrey F. Naughton, and Philip A. Bernstein, editors, *ACM SIGMOD International Conference on Management of Data*, page 601, New York, 2000. ACM Press.

[24] P. Revesz and S. Wu. Visualization of recursively defined concepts. In *Proceedings of 8th International Conference on Information Visualization*, pages 613–621, London, 2004. IEEE Computer Society.

[25] D. Shepard. A two-dimensional interpolation function for irregularly spaced data. In *Proceedings of the 23nd ACM National Conference*, page 517524, New York, 1968. ACM Press. Chairman-Richard B. Blue, Sr. and Chairman-Arthur M. Rosenberg.

[26] A. Silberschatz, H. Korth, and S. Sudarshan. *Database System Concepts - 5th edition.* McGraw-Hill, New York, 2005.

[27] U.S. Geological Survey. West Nile virus maps at http://westnilemaps.usgs.gov/, (Accessed: 12 December, 2004).

[28] C. N. Theophilides, S. C. Ahearn, S. Grady, and M. Merlino. Identifying west nile virus risk areas: The dynamic continuous-area space-time system. *American Journal of Epidemiology*, 157(9):843–854, 2003.

[29] M. F. Worboys. *GIS: A Computing Perspective.* Taylor & Francis, London, 1995.