Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia
Social and Behavioral Sciences

Procedia - Social and Behavioral Sciences 73 (2013) 410 – 417

The 2nd International Conference on Integrated Information

Investigative strategies on top of a spatio-temporal database about sex offenders

Paolino Di Felice*

Dipartimento di Ingegneria Industriale e dell'Informazione ed Economia, Università di L'Aquila (Italy)

Abstract

To combat against persistent sex offenders, recent laws call for the use of the GPS technology to monitor their movements. In [1], it has been proposed the adoption of a Spatio-Temporal DataBase (S-T DB) for archiving this type of complex data, besides traditional ones (i.e., data about their home, (pending) crimes, and sensible areas). In the present paper we discuss investigative strategies that can be easily implemented on top of such a kind of database by taking advantage of the history of the movements of sex offenders. As implementation platform, the SECONDO DB Management System (DBMS) is adopted since it supports a data type about moving objects as well as a reach set of spatio-temporal operators to query such complex data [2].

Selection and peer-review under responsibility of The 2nd International Conference on Integrated Information

Keywords: sex offender; criminal mobility; moving points; database; data analysis; querying.

1. Introduction

Fight against criminals is a tremendous challenge that must be addressed using any lawful method or tool, because the stake as a prize is the safety of citizens. In this demanding scenario, from many years have acquired an important role software developed to support the investigators' job (useful information on the subject can be found, for example, in [3] and [4]). At a high level of abstraction, investigators need methods of inquiry that support them trying to:

- limit the list of potential culprits from which to start the investigation, downstream of an abuse (in short, *punitive* action),
- prevent the frequent episodes of recidivism among the serial sex offenders registered in the DBs they have access to, through the periodic analysis of their movements (in short, *predictive* action).

* Corresponding author: Paolino Di Felice. Tel.: +39-0862-434 418; fax: +39-0862-434 403.
E-mail address: paolino.difelice@univaq.it

Selection and peer-review under responsibility of The 2nd International Conference on Integrated Information
doi:10.1016/j.sbspro.2013.02.069

Most of the software today in use interact with traditional DBs (i.e., DBs that collect data modeled as standard data types – **`string`**, **`int`**, **`float`**, **`date`**, …) spatially extended (i.e., hosting geometric attributes of type **`point`**, **`line`** and **`region`**, essential to describe, for example, the geographic location of the criminals' home as well as the location where sex offences took place, besides streets, sensible areas, and so on – *CrimStat*, for instance, belongs to this category [5]).

The relatively recent advent of laws that authorize much more severe forms of control, such as those based on the monitoring h24 of sex offenders through the GPS method, reopens the discussion about what should be the best technological solution to be adopted. In particular, it is to be decided *how* to treat the data about their movements.

In [1] (a paper that precedes and completes the present one), it has been argued that the time is ripe to adopt a DBMS of a new generation able to support data types and operators for the so-called moving points (in short, *m-points*) ([6] is an excellent textbook on the subject). Aim of the present paper is to give a contribution to this respect. As DBMS, we will make use of SECONDO.

The article is organized as follows. Sec. 2 recalls the application context our study refers to; the logical schema of the DB concerning (potential) crimes, sensible areas, subjects on probation, and their trips between pairs of areas; and, lastly, the example dataset used to carry out a few preliminary experiments. Sec. 3 focuses on possible investigative strategies both punitive and predictive, showing their implementation in SECONDO, as well. Sec.4 concludes the paper.

2. The application context, the database, and the example dataset

The reference scenario of the study reported in this paper relates to sex offenders subject to the constraint of surveillance in their movements (hereinafter, therefore, also called *subjects on probation*), sensible areas (schools, parks, public restrooms, train stations, ...) and movements of subjects on probation (acquired by means of an electronic device equipped with a GPS detector) between sensible areas.

Throughout the paper we use the notations $\mathbb{C}$, $\mathbb{P}$, $\mathbb{S}$, $\mathfrak{A}$, and $\mathbb{T}$ defined as follows. Let $\mathbb{C}=\{C_1, C_2, \ldots, C_c\}$ denote the known sexual crimes; $\mathbb{P}=\{P_1, P_2, \ldots, P_p\}$ the pending sexual crimes; $\mathbb{S}=\{S_1, S_2, \ldots, S_s\}$ the subjects on probation, $\mathfrak{A}=\{A_1, A_2, \ldots, A_a\}$ the sensible areas, and $\mathbb{T}=\{T_1, T_2, \ldots, T_t\}$ the movements. In turn, T_i expresses the trip of a subject in $\mathbb{S}$ between two sensible areas in $\mathfrak{A}$. Formally, T_i = {<P,t> | *P* is a point described by a pair of coordinates <lat, long> and *t* is the time stamp of the acquisition of *P*}, where i=1, 2, ..., $|\mathbb{T}|$. The elements of T_i are temporally ordered.

The values from the sets $\mathbb{C}$, $\mathbb{P}$, $\mathbb{S}$, $\mathfrak{A}$, and $\mathbb{T}$ can be stored, in sequence, in a SECONDO S-T DB structured in terms of the following six tables:

```
let crime = [const rel(tuple ([IDCrime: string, Description: string ])) value ()]
let pendingCrime = [const rel(tuple([IDPC: string, When: instant, Where: point,
    Description: string, Against: string])) value ()]
let criminal = [const rel(tuple ([IDCriminal: string, Name: string, Home: point]))
    value ()]
let sensibleArea = [const rel(tuple ([IDSA: string, Type: string, City: string,
    State: string, Position: point, Layout: region ])) value ()]
trip (IDCriminal: string, TripData: mpoint, From: string, To: string)
let perpetrate = [const rel(tuple ([IDCriminal: string, IDCrime: string,
    When: instant, Where: point, Against: string ])) value ()]
```

Remarks about this solution are given in [1].

To make a few preliminary experiments about the investigative strategies that will be discussed in the next section, we refer to the synthetic dataset introduced in [1]. It consists of 20 subjects on probation, 20 sensible areas (located in the town of L'Aquila and nearby – centre of Italy – and composed of 7 elementary schools, 6 middle schools, 4 public restrooms, and 3 parks), and 30 trajectories corresponding to as many trips of the 20 subjects, between pairs of sensible areas.

3. Examples of investigative strategies

In the following we propose *two* + *one* investigative methods that implement, in sequence, the punitive action and the predictive action. The final goal is to show how easily they can be implemented in SECONDO. Notice that, queries can be formulated to SECONDO either in an SQL-like language (top level) or in the executable language (low level). Hereinafter, we use this latter way.

3.1. Punitive action

We will discuss two alternative tests. The first one is based on theoretical researches and experimental validations well-known in the *Environmental criminology* domain [7], while the second test is new.

Test A

Investigation requirement

The usual way to start up an investigation, immediately after an episode of sexual violence, is to verify the alibi of all subjects on probation living in relatively close proximity to the area of crime (*where*). The rationale behind this action is based on the so called *distance-decay pattern*, [8].

Input-output of Test A

INPUT

$\mathbb{S}$, *where* (the place of the abuse), d (a distance value)

OUTPUT

$\mathbb{S}_{where}{}^{d} \subseteq \mathbb{S}$ (i.e., the list of criminals whose home is at a distance less than d from *where*. The list is returned in ascending order of the value of the distance from *where*)

Implementation

It is accomplished in three steps discussed below in sequence.

a) Creation of a buffer around the crime scene.

```
let areaCrime = pendingCrime feed extend [Area:circle (.Where, 0.02, 50)] consume
```

The query constructs, using the **`let`** command, the object **`areaCrime`** in two steps, as follows. The **`feed`** operation (applied in postfix notation) reads the **`pendingCrime`** relation from disk and puts its tuples into a stream. **`extend`** adds the attribute **`Area`** to the current tuple of the mentioned stream. The value of such an attribute is computed by the **`circle`** operator which receives as input the center of the buffer (.**`Where`**), the radius (2km – value modifiable at will, but close to the value that domain studies deem significant, [8]), and the number of sides (50) of the regular polygon which approximates the circumference. Lastly, the **`areaCrime`** stream of tuples is transformed into a persistent relation by the **`consume`** operation.

b) Identification of criminals residing inside the buffer.

```
let nearbyCriminals =
   criminal feed {cr} areaCrime feed {ac} symmjoin[.Home_cr inside ..Area_ac]
   renameattr[IDCriminal:IDCriminal_cr, Name:Name_cr, Home:Home_cr,
           IDPC:IDPC_ac, Area:Area_ac, Where:Where_ac]
   project [IDCriminal, Name, Home, IDPC, Where, Area]
consume
```

A crucial operation is the symmetric join between tables **`criminal`** and **`areaCrime`** (**`cr`** and **`ac`** denote, in sequence, their aliases) that performs a Cartesian product from its argument streams and filters according to the join condition that involves the **`inside`** spatial operator. (In the abbreviated form used above: "." refers to the first argument of the join, while ".." denotes the second argument. More specifically, **`.Home_cr`** denotes the attribute **`Home`** of table **`criminal`**, while **`..Area_ac`** denotes the attribute **`Area`** of table **`areaCrime`**.)

c) Sorting of suspects according to the value of the distance that separates their home from the crime scene.

```
query nearbyCriminals feed
   projectextend [IDCriminal, Name, Home, IDPC, Where;
         Distance: real2int((distance(.Home,.Where)) * 100000)]
   sortby [Distance asc]
consume
```

(The keyword **`query`** starts queries written in the SECONDO's language.)

The result of *Test A* against the example S-T DB is shown in Fig. 1.

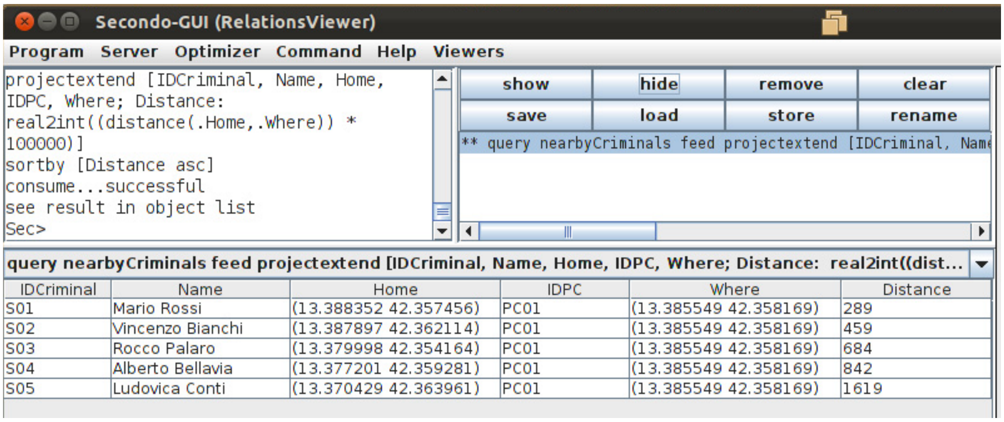

Figure 1. Distance (in meters) of the place of the sexual abuse from the home of the offenders under investigation.

Test B

This is a new test, which is not rooted into the environmental criminology literature, yet it seems appropriate to be performed because of its generality, in fact, it applies both in the case of a single episode of sexual abuse than serial. This test is encouraged by the existence of an ST-DB of the type proposed in this article that, as we shall see, makes the implementation trivial.

Investigation requirement

The idea is the following. Downstream of an episode of sexual violence, a rapid check to be carried out by querying the S-T DB is to verify, for each subject on probation stored in it, whether he/she have made a movement that brought him/her to pass near by the crime scene (*where*) the same date of the crime and in an hour close to that of the abuse (*when*). If the test result was positive, it would be a very strong evidence against the subject.

Input-output of Test B

INPUT

$\mathbb{S}$, $\mathbb{T}$, *where* (the place of the abuse), *when* (date and time of the abuse)

OUTPUT

$\mathbb{S}_{where}^{when} \subseteq \mathbb{S}$ (*i.e., the list of criminals* for which in the S-T DB is stored a movement that saw them be in proximity of *where* in a time stamp close to *when* – i.e., same date of the abuse, and time close).

Implementation

It is accomplished in two steps discussed below in sequence.

a) build a buffer around the crime scene. It is necessary to have recourse to a buffer to absorb all the causes of uncertainty about the exact location of the criminal (alias m-point) [9]. The buffer radius is sufficient to be of some tens of meters:

```
let buffer = pendingCrime feed extend [Area:circle (.Where, 0.0001, 50)] consume
```

b) process the movements of each sex offender in the S-T DB which began before the abuse and ended just after, in order to assess whether he/she entered the area covered by the buffer in a moment close to that of the crime.

```
query trip feed {tr} buffer feed {bf}
   symmjoin[.TripData_tr passes ..Area_bf]
   extend [Range:theRange(inst(initial(.TripData_tr at .Area_bf)),
           inst(final(.TripData_tr at .Area_bf)), FALSE, TRUE),
           Trajectory: trajectory(.TripData_tr)]
   filter [.Range intersects theRange(
           .When_bf - [const duration value (0 3600000)],
           .When_bf + [const duration value (0 3600000)], FALSE, TRUE) ]
   renameattr [IDCriminal:IDCriminal_tr, IDPC:IDPC_bf, Area:Area_bf, When:When_bf]
   project [IDCriminal, IDPC, Area, When, Trajectory]
consume
```

The operator **passes** ascertains which of the subjects on probation registered in the S-T DB entered the computed buffer area and, for each of them, the time of entry and exit from such an area is calculated (operators **inst(initial())** and **inst(final())** applied to the m-point (attribute **tripData**)). Then, **theRange()** builds the corresponding time window. To see the path of the sex offender, the output of the query is extended with the attribute **Trajectory** that contains the trajectory of the m-point, calculated by applying the operator **trajectory** to the field **tripData** of type **mpoint**. Through the unary operator **filter** which, in turn, uses the **intersect** operator, the suspects are shrank to only the sex offenders who have passed through the buffer in a temporal window between one hour before and one hour after that of the abuse (*When*).

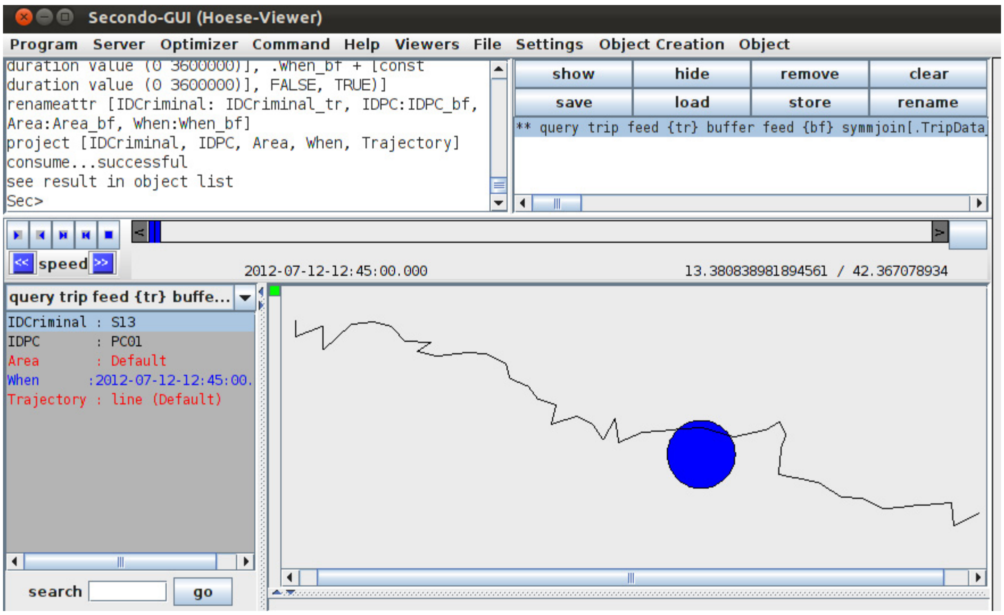

Figure 2. Output of *Test B*. *Default* is a SECONDO's category of visualization of lines.

Fig. 2 shows the result of *Test B* applied to the example S-T DB; i.e., there is only one trajectory (ascribable to the subject on probation *S13*) that crosses the buffer (**`Area`**) built around the crime scene (*Where*), this latter identified by code IDPC = "PC01".

3.2. Predictive action

The secret wish of any civilized society is to defeat crime. Having a S-T DB of the type of Sec.2, it allows us to carry out investigations about the movements of the subjects of probation in order to recognize patterns of dangerousness. Therefore, it is our opinion that putting it in exercise is one of the prerequisites to achieve this ambitious goal. Below we discuss a simple investigation of real interest. It is just an example, but it allows to state a general concept: should any new need arise, to satisfy it will be sufficient to query the DB, provided that the underlying DBMS features spatio-temporal operators.

Test C

Investigation requirement

To know the length of stay (measured in minutes) of each criminal inside the sensible areas in a given temporal window (measured in days, weeks or months), so that to be able to carry out selective checks for each "anomalous" stop (for example, the permanence of one hour in a public restroom).

Input-output of Test C

INPUT

$\mathbb{S}$, $\mathbb{T}$, $\mathbb{A}$, *interval* (a temporal window, e.g., 1 month)

OUTPUT

$\mathbb{S}_{\mathbb{A}}^{interval} \subseteq \mathbb{S}$ (*i.e., the list of* subjects of probation for which in the S-T DB is stored a movement that saw them remain inside a certain sensible area for a time interval at least equal to *interval*.)

Implementation

It is accomplished in three steps discussed below in sequence.

a) Detection of the criminals' movements.

```
let tripLastMonth = trip feed filter
    [inst(initial(.TripData))>(today()-[const duration value (30 0)])]
consume
```

The object **`tripLastMonth`** contains the movements carried out by each subject of probation in the last month (value modifiable as desired).

b) Creation of table **`stay`** which puts together criminals and the entry – exit interval from the sensible areas.

```
let stay = tripLastMonth feed {trip1} tripLastMonth feed {trip2}
              symmjoin[(.From_trip1 = ..To_trip2) and
                   (.IDCriminal_trip1 = ..IDCriminal_trip2)]
    renameattr [IDCriminal:IDCriminal_trip1, SensibleArea:From_trip1,
              Trip1:TripData_trip1, Trip2:TripData_trip2]
    projectextend [IDCriminal, SensibleArea;
              ComeIn:inst(initial(.Trip2)), ComeOut:inst(final(.Trip1))]
consume
```

c) Creation of table **`stayPeriod`** which adds to the attributes of table **`stay`** another attribute (**`Period`**) containing the duration (in minutes) of the permanence of a sex offenders inside a sensible area.

```
let stayPeriod = stay feed
    extend[Period: real2int((millisecond_of(.ComeOut - .ComeIn))/60000)]
    sortby [Period desc]
consume
```

d) Detection of stays longer than a threshold value (e.g., one hour).

```
query stayPeriod feed filter [.Period > 60] consume
```

The output of *Test C* is omitted because of space limits.

3.3. Summary

Table 1 puts together the three tests discussed, for each of them showing the category of offence it applies to (single/serial offence, both cases), the enabling DBMS technology (i.e., DBMS either spatially extended or spatio-temporal extended), the action it refers to (either punitive or predictive).

Table 1. The discussed tests at glance

Test	Scope	DBMS	Action
A	Single offence	Spatially extended	Punitive
B	Both cases	Spatio-temporal extended	Punitive
C	Both cases	Spatio-temporal extended	Predictive

4. Conclusions and future work

The author has given his point of view about a technological innovation that once introduced could offer a much better support in the investigation against persistent sex offenders: the transition from DBMSs with spatial extension to those for moving objects. These latter category of systems will make possible to put in exercise DBs about (potential) crimes, sensible areas, subjects on probation, and their trips between pairs of areas, to be equipped with a package of ad hoc queries that implement the investigations of interest.

This is an effective and efficient solution, with low costs of implementation and operation, open with respect to any investigative requirement that may arise over time, in fact to face each new need it is sufficient to write an ad hoc query. In particular, the effectiveness of the proposed solution is destined to raise over time with:

- the growth of the DB size in terms of subjects on probation and their movements,
- the growth of the number of events of recidivism charged to them,
- the identification of an appropriate number of "comparison parameters", whose values act as a reference threshold to be taken into account to interpret the movements of the criminals. In a nutshell, the message is that to write queries against the DB is not enough, what has to be done first is to equip ourselves with "behavioral parameters of reference" for each category of crime and for each sex offender's profile, whose "threshold values" have to be updated periodically querying the DB.

Last but not least, time should be spent to understand the many implications deriving from the availability of a type of data much richer than that ever used before. In particular, new investigative algorithms, that take benefit from the knowledge of the full history of the movements of the sex offenders as well as of their past history (places of sexual abuses and modus operandi), should be looked for.

The writer is convinced that the adoption of such a technological solution opens the frontier to a new generation of software applications much more effective than those currently used by several criminal investigation departments all over the world. The concrete benefits of this innovation will fall on the entire community that will have the chance to live in safer cities.

Acknowledgements

Many thanks to students Luca Di Lonardo and Maurizio Petrocco who carried out the implementation.

References

[1] Di Felice, P. (2012). The design of a spatio-temporal database to investigate on sex offenders. Proc. 2nd Inter. Conf. on Integrated Information.
[2] Güting R.H., Behr, T. and Düntgen, C. (2010). SECONDO: A platform for moving objects database research and for publishing and integrating research implementations. IEEE Data Engineering Bulletin 33:2, 56-63.
[3] Kemshall, H. (2001) Risk Assessment and Management of Known Sexual and Violent Offenders: A review of current issues. Police Research Series -- Paper 140.
[4] Rich, T., Shively, M (2004). A Methodology for Evaluating Geographic Profiling Software. Document No. 208993.
[5] http://www.icpsr.umich.edu/CrimeStat/
[6] Güting, R.H., Schneider, M (2005). Moving Objects Databases. Morgan Kaufmann Publishers.
[7] Brantingham, P. J., Brantingham, P. L. (1991). Environmental criminology. Prospect Heights, IL: Waveland Press.
[8] Beauregard E., Proulx, J., and Rossmo, D. K. (2005). Spatial patterns of sex offenders: Theoretical, empirical, and practical issues. Aggression and Violent Behavior 10 579–603.
[9] Trajcevsky, G., Wolfson, O., Hinrichs, K., and Chamberlain, S. (2004). Managing uncertainty in moving object databases. ACM Transactions on Database Systems, 29(3), 463-587.