Spatial and temporal information structuring for natural risk monitoring

Guillaume Noël, Sylvie Servigne, Robert Laurini

Spatial and Temporal Information Structuring for Natural Risk Monitoring

II International Conference and Exhibition on Geographic Information, Estoril Congress Center, May 30- June 2, 2005

Guillaume NOEL, Sylvie SERVIGNE and Robert LAURINI

The work presented in this paper concerns Global Environment, focusing on Natural Risk Monitoring: our application is volcanic activity monitoring.

The real-time data used in such a system generally comes from an array of sensors and are usually collected into a central database. With the now widespread addition of GPS, these data therefore have both temporal and spatial components.

Therefore a major trouble encountered is the data structuring, so that it can store data effectively and process real-time queries on real-time objects.

Our work offers a new indexing method dealing with multi-dimensional aspects of the data as well as storing and real-time querying.

KEYWORDS

Spatio-temporal, Real-time, Natural disaster monitoring, Database indexing

INTRODUCTION

More and more natural environment monitoring systems rely on small sensors disseminated within the environment to study. These sensors produce data that must be collected with real-time constraints into a database. The adjunction of GPS enabled sensors adds a spatial component to the temporal one in the definition of the data. A data is generated from a sensor and can be qualified by the spatial location of this sensor as well as by the timestamp associated with the measurement date. The database storing the data has to take into account these different properties, even more so in its access structure. The work presented in this paper concerns Global Environment Monitoring, more precisely Natural Risk Monitoring: our application is volcanic activity monitoring [2]

Different indicators assess the risk level (seismic activity, gases components, slopes clinometry,...). They can be measured thanks to an array of geographically referenced sensors. They later on send the data to a centralized database. Our focus is the study of methods for an efficient structuring of the gathered data. These data have specific properties: they are multidimensional as well as real-time.

This paper is divided into four parts. The first one presents different characteristic of the data collected by such sensor data, more precisely the multidimensional and real-time aspects of the data. On the second part is described the volcanic activity monitoring process. From this, the third part details the PoTree, a spatio-temporal sensor based database indexing structure. Then, the fourth part consists of a description of the PasTree, another indexing structure aiming at widening the querying capacities, thanks to a more precise definition of the data dimensions.

1. SENSOR DATA PROPERTIES

The use of sensors in monitoring systems, for natural or industrial phenomenon, implies different properties [3] [6]. While the constant flow stemming from periodic measurements adds real-time constraints, the nature of the data adds multidimensional properties.

1.1 Multidimensional data

Multidimensional data imply in this case spatial, temporal and semantic properties.

As the sensors are spatially referenced, it is possible to link every measurement to a specific location. In volcanic activity monitoring, measurements are linked to a 3D mapping of the volcano. Not all data stored within a monitoring system database are directly linked to measurements. As a matter of fact, aggregation of measurements can lead to further data. Triangulation is a common method for determining the true position where an event occurred. Passive sensors (sensors that record the data emitted by an event without probing the system) are usually used in such a context.

- Just as the data are linked to a sensor, they also bear timestamps notifying the measurement date. Therefore they are also temporally defined. Data from a specific sensor can be ordered through their respective timestamps. It is to be noted that sensors can have different modes of measurement. While some sensors have strictly periodic measurements, others only send trap-like alerts when a threshold is reached. Therefore, while the timestamps of the periodic measurements can be predicted accurately under normal circumstances, event-driven sensor measurements only happen during specific activity phases. The timestamp associated to these data are therefore particularly important so as to determine the evolution of the studied system.

- Finally, data come from different kind of sensors, providing more or less important data to users, or they can also stem from calculations based on measurements (e.g. epicenters localization derived from the data issued by seismographs). Therefore it is possible to define semantic properties linked to the data. More and more researches are dedicated toward the semantic properties of the data [14]. The main idea is to gather semantically similar data into specific clusters. This solution aims at accelerating query resolutions by directly forwarding the queries toward the corresponding clusters. Furthermore, it has been notified that queries that are emitted by the same user tend to access the same kind of data. Therefore, semantic storing of the data within the system can save query-resolving time. However, these solutions are usually linked to the use of clusters, distributed databases and replication. These properties usually appear as troublesome when the updating scheme imply soft real-time properties. They usually are synonyms of longer processing and transmission time, leading to more transactions not being able to respect their real-time constraints.

From all of these properties, it can be noted that the data stemming from sensor networks offer a wide array of multidimensional properties. While they are inherently spatially referenced, they also bear temporal properties. Finally, the variety of sensors leads to different semantic properties as well. The database managing these data can use some of all of these properties as access methods. However, while these factors are important, the data updating rate also bears much weigh in the database management system design. While some sensor networks only provide few periodic data, an efficient system must be able to cope with heavy workloads of updates, periodic or aperiodic, corresponding to specific activity phases. Therefore, while some systems focus on query resolution, a sensor database must not underestimate the real-time updating constraints.

1.2 Real-time

As stated above, the use of a sensor network implies real-time constraints. As a matter of fact, measurements must be stored and indexed before a newer version reaches the central database [13].

From this statement, we can provide a more precise definition of real-time. In computer sciences, real-time means “time constraints respect”. In a database, this means that transactions bear a time limit. They must be answered before the end of this limit. Usually, it is recognized that no answers to a query is better than a late answer, which could bear outdated data.

It is possible to distinguish three real-time constraints:

Soft real-time imply that the query should be answered before their deadlines, but eventually, it could be acceptable that some queries cannot meet this requirement.

Firm real-time imply that even though it is still possible for some queries to be unable to meet their real-time requirements, this tardiness can lead strongly negative results for the system.

Strong real-time implies that under no circumstances a transaction should be allowed to miss its deadline. This could lead to a critical system failure.

In most systems, real-time is often associated with the notion of priority. A real-time transaction has a priority according to the importance of the transaction. Different priority assignation methods exists (rate monotonic, earliest deadline first,… [8]). The management system then uses these priorities to schedule a computing solution that aims at respecting every time constraints. It should be noted that while some transactions or data have real-time properties, others might not be real-time.

In sensor networks, the measurement rate implies that the data collection and storing system must be able to cope with the flow of gathered data. Therefore, it is possible to recognize real-time properties linked to database updating. For periodic measurement systems, it is usually assumed that the evolution of the measurement values is more interesting than the values. As such, it is usually admissible to lose some measurement values, if these values do not show drastic evolution of the system. Such a concept has lead to the notion of epsilon-data, where data updating can be skipped when only minor changes are reported between two versions.

For aperiodic measurement systems (systems based on thresholds), the lost of one measurement can be more significant. However, sensors usually use redundancy to cope with the lost of one specific sensor. If a given sensor disappears from the network, other sensors should be able to replace it. Furthermore, as measurement device configuration can drift through time (becoming less and less accurate), sensor redundancy allows for some data validity security.

Therefore, it appears that some data updating could be skipped under normal circumstances. Therefore, the data updating method should aim at soft real-time constraints.

There is also another real-time factor to consider. Actually, Environment Monitoring, and even more so Natural Risk Monitoring aims at providing tools for decision support. The pertinent data should be provided to the right people on time. Under normal circumstances, scientist and users as a whole accept some data processing delay before accessing the data they asked. Such data-accessing scheme is not real-time in itself. However, in times of crisis, pertinent data should be obtained within time constraints, so as to allow fast decision making and pertinent crisis response. Such data-accessing scheme is real-time as it requires the data to be accessible within time constraints.

As a partial conclusion, it should be noted that the data issued by a sensor network bear real-time properties. The data measurement flow is source to soft real-time constraints. As for data accessing, normal circumstances does not imply any real-time properties. However, during crisis, data should be available to the decision makers within time limits. This also means real-time… Therefore, sensor related data are inherently real-time, and data from Natural Disaster Management systems does comply with these properties.

2 VOLCANIC ACTIVITY MONITORING

Volcanic activity monitoring relies on the use of heterogeneous types of sensors, dispatched in different stations. While satellite imagery provides post-eruption data, day to day monitoring is usually conducted through sensor networks.

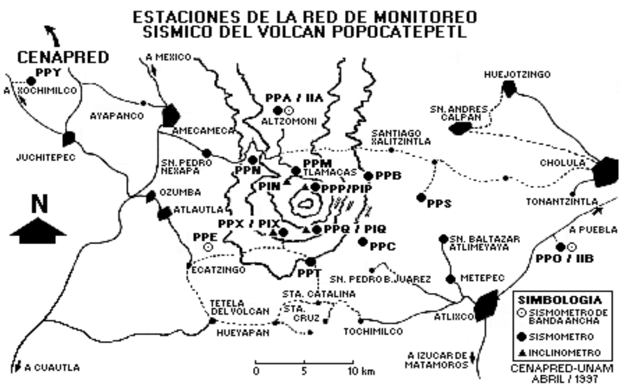

Figure 1: Sensor network monitoring the activity of the Popocatepetl

The Cenapred has been commissioned by the Mexican government to manage the data coming from the Popocatepetl monitoring network. The Popocatepetl is a volcano located between Mexico City and the city Puebla, and spans on three different states. An array of stations has been set up to gather the data from the volcano, as seen in figure 1. The sensors vary from different seismographs to clinometers.

One important fact in Natural Risk Monitoring is the potentially low lifetime of a sensor. Usually located in dangerous zones, under harsh meteorological conditions, sensor faults are to be expected [15]. Because of this phenomenon, the data measured by the sensors must be regularly sent to a central location. The data transmission policy must take into account the cost of transmission as well as the necessity to maintain a consistent database.

The data are collected into a central database and processed accordingly. The current database uses the Earthworm system provided by the USGS [19]. The data are currently indexed according to the sensor identifier and the measurement timestamp. Spatial properties are seldom used. However, with the evolution of agile sensor networking (sensors that can change position), the spatial component of the measured data is becoming an important factor, as important as the temporal one.

Usually specialists tend to admit that the most recent data is the most interesting, apart from data concerning specific activity phases. Analyzing process to recognize the current volcanic activity phase through pattern matching can use such data.

Due to continuous acquisition of measurements, the database has to use mechanisms emphasizing the importance of the newest data. This is even truer for indexing methods. While traditional querying uses sensors identifiers, scientists now ask for the ability to use spatio-temporal queries. They wish to find the data issued during a specific time interval within a spatial zone.

To sum it up, volcanic monitoring systems use the most recent data to determine the current state of the volcano, through pattern matching. While traditional query processing makes use of sensor identifiers, the development made in agile sensor networks now gives a new importance to the spatial dimension. This aspect must be included into the database management system.

We have decided to focus our work on indexing methods for spatio-temporal data in a real-time environment. Many studies have already dealt with real-time constraints [8] [10] or spatio-temporal indexing [9] [12], however there appear to be little work on real-time spatio-temporal indexing. Therefore, we propose solutions.

3. FIRST PROPOSITION THE POTREE

The PoTree [11] has been designed for real-time spatio-temporal updating and real-time spatio-temporal querying. It does not focus on the semantic aspects of the data but gathers the data from a specific sensor into specific subtrees. More specifically, it divides the global dataset through different subtrees, as seen in figure 2. A global spatial subtree acts as the root of the PoTree. Each of its leaves points to a temporal, sensor specific subtree. The spatial subtree is first used to determine which temporal subtree must be queried.

3.1 PoTree structure

Contrary to the approaches based on R-tree [4], the most commonly found indexing solution in actual spatio-temporal databases, the PoTree separate temporal and spatial dimensions. The specification of sensor network based data allow for such separation. As a matter of fact, current measurement sensors are mainly fixed, with limited additions of new stations.

The number of sources of information (sensors) remains restricted in our case. The queries are usually linked to spatial and temporal components: “what are the data gathered in the zone defined by the rectangle <X1,Y1 ; X2,Y2>?” As the data are gathered from spatially referenced, fixed sensors, the spatial component of the query can be used as the most restrictive factor, therefore the most useful factor for limiting the data search. Queries first have to determine which sensor has emitted the queried data and from this find the data from the sensor through temporal search. This method limits the portion of data to be searched through in order to answer the query.

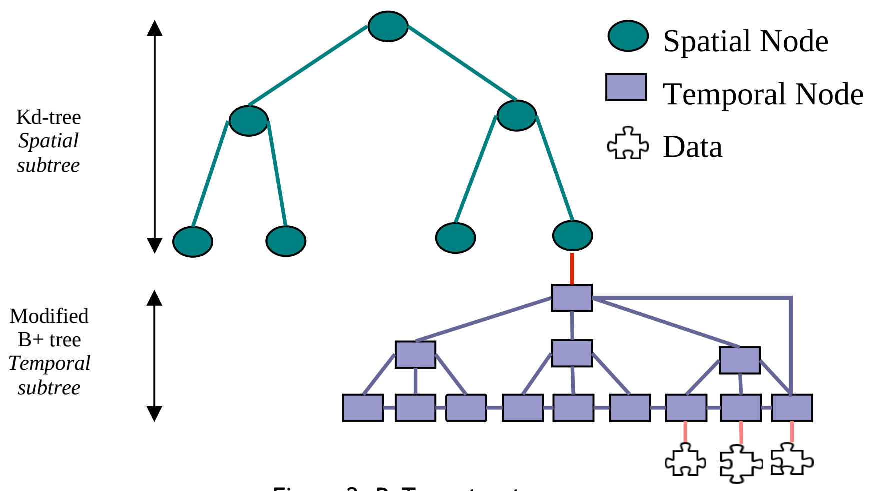

Figure 2: PoTree structure

The data emitted from a specific sensor must be added incrementally and in real-time. Linearization of the data proves to be helpful on this aspect.

The PoTree uses a kd-tree [1] for the management of the spatial component of the data and modified B+ trees (actually closer to an AP trees) for the temporal one. On these subtrees, as the updates are incremental, new entries shall always be made on the rightmost leaf node.

The choice of the kd-tree is justified by comparisons made with the R tree. As a result of these comparisons, it appeared that the kd-tree offers flexibility and fast build time when compared to the R-tree. Furthermore, the kd-tree can be updated on-line (addition of new sensors can be made while the system is running), which is harder to achieve with R-trees. Nonetheless, the properties of the kd-tree induce the fact that the shape of the spatial subtree of the PoTree is directly linked to the spatial data insertion order, therefore it may be unbalanced. As sensor additions are limited, this should not prove to be a crucial factor.

The use of B+-tree to index temporal data (using the data timestamps) is justified by the possibilities offered as regards to update and other queries. Moreover, this structure could be developed in order to facilitate the integration of the database management system into a datawarehouse. The B+ tree has been modified to include a direct link between the root of the tree and the rightmost leaf node, the node where the most recent data shall be found. Therefore, a query within these substructures shall first search if the queried data are located within the last node. In that case, the query shall directly access this node, without crossing the different internal nodes of the subtree, thus saving processing time.

The use of a dual structure makes it possible to associate each spatial localisation to a temporal subtree. This way, each of these subtrees represents the data stemming from a source of information, a specific fixed sensor.

As it appears on the diagram (figure 2), the leaves of the spatial subtree (kd-tree) point are towards temporal subtrees (B+trees). As for this subtree, the leaves point to the actual data. The nodes of the kd-tree comprise only data relating to the spatial position of the objects. They do not carry information relating to the temporal dimension. As for the temporal subtrees, the nodes of B-trees do not carry information relating to the geographical position of the objects, but only information relating to time. While this feature limits data duplication, it also limits the possibilities of mobility management. As sensor agility is becoming a reality, this could prove to be limitation of the structure if mobility is aimed at.

3.2 Structure tests

Comparisons has been carried out between the PoTree and R*trees, as these kind of trees as regularly encountered in spatio-temporal databases. Different tests have used both computer generated data and real seismic data collected from the Japanese K-net [7].

The comparisons have been carried out on a computer running under Linux, with a processor at 1.6 GHz and 128 MB of RAM. The language used was Java. The R* tree source code used was the one provided by Hadjieleftheriou [5]. The tests focused on the tree construction time of the trees and on spatial-windows / temporal-interval queries, the most common queries.

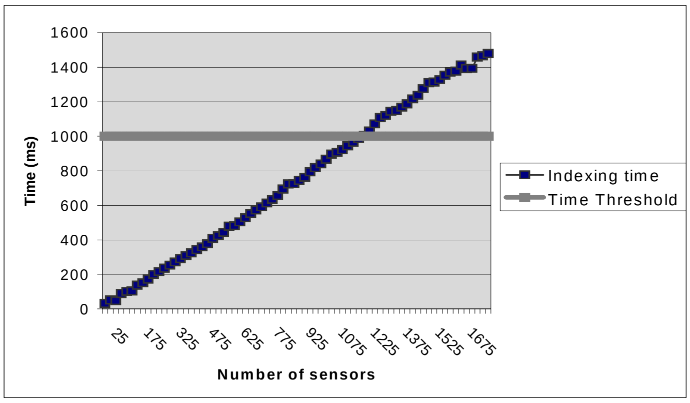

Figure 3: influence of the number of fixed sensors on the construction time of the PoTree

The first test focused on the tree construction time. The figure 3 shows the evolution of the construction time according to the number of different sensor to index. The figure shows the equivalent of 1 second of computer-generated data (seismographs with a frequency of 100 Hz). As shown on the figure, it is possible to index data coming from up to 1200 different sensors while respecting the real-time updating constraints. However, on a real-system, other queries, such as continuous queries would lower this number.

The PoTree divides the data according to the spatial references of the sensors. For the temporal dimension, the updates are made incrementally. Therefore, under normal circumstances, the temporal subtrees take the full advantage of the direct link between the root and the last leaf node.

Therefore, it can be assumed that the global construction time is $O(\sum_{1}^{i} N_i(\ln(N_i+1)))$ where *i* is the number of different spatial position used and N_i is the number of data issued from the position *i*.

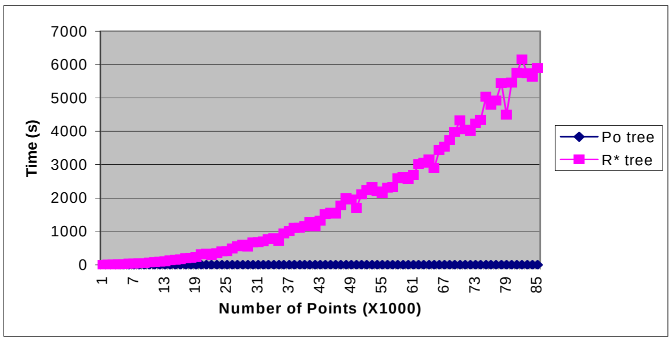

Figure 4: influence of the number of sensors on the construction time of the PoTree and R* tree

The figure 4 shows the construction times of the PoTree and R*-trees. As the R*-tree doesn't differentiate the spatial and temporal dimensions, and as updates needs to take into account the whole dataset, the construction time for such a tree can lead to a situation where it becomes impossible to answer the real-time constraints. On the other hand, the Po-tree subdivides the dataset. It uses different temporal trees to index the data, according to the sensors they are linked to. Figure 5 shows the construction times for both of these structures. It appears that the Po-tree construction time is far less, as the number of spatial locations is limited and temporal sub-trees only needs to access the root and last nodes, with the exception of the B+ tree splits when a node is filled.

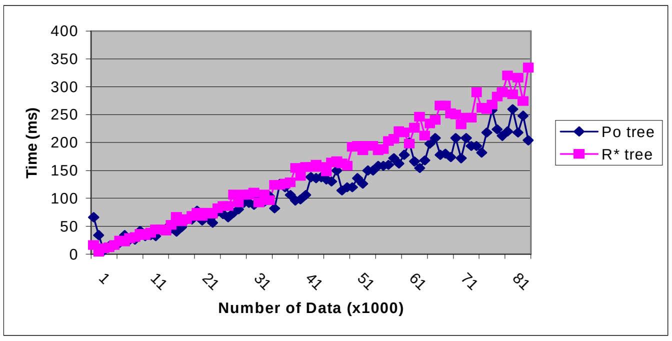

Figure 5: influence of the number of data on spatial windows / temporal interval queries (last 10% of the entered data)

Other queries have shown interesting properties as well. The interval queries took an advantage of the linking of the temporal nodes of the Po-tree. For point-interval queries, the Po-tree can be up to 8 times faster the R*-tree. While for windows-interval queries the difference has shown much lighter, it still remains in favour of our solution, as shown in figure 5. On this figure, the last 10% of the entered data where fetched. The spatial range covered the whole possible locations. For a low number of data the R*-tree fares better, yet when the amount of data rises past 6000, it is the Po-tree that gains the advantage. These results can be explained by the fact that R*-trees need to consider the whole set of data so as to perform a query. Therefore, the more data are indexed, the longer it takes to find specific values.

Tests with real data (indexing an earthquake followed by 68 seismographs at 100 Hz) have confirmed the results from the computer-generated data.

The PoTree have proven an effective solution for indexing data from a network of fixed sensors. The structure, sensor oriented, gives satisfying results. However, the advances in agile sensor technologies now call for the development of structures able to cope with sensor position changes. Furthermore, different data processing activity can also generate new kind of data (spatio-temporal aggregates, [20]), which can be considered as sensor related data to some extend. For example, seism epicentre localisation can be determined from triangulation. The epicentres from different seism can be considered as data issued from an agile sensor. The PoTree cannot manage this kind of data efficiently. Thus the Pastree specifications were developed…

4 SECOND PROPOSITION: THE PASTREE

An evolution of the PoTree is the current development of the PasTree. This structure widens the possibilities of the PoTree by blurring the dimensional separation between subtrees. Changing the spatial subtree to include multiversion properties enables indexing of sensors with low to medium agility. Adding a differentiation between spatially similar updates and position changing updates allows for sensor specific queries. A dedicated subtree can be introduced to refer the sensors through their identifiers to allow for a wider range of possible queries.

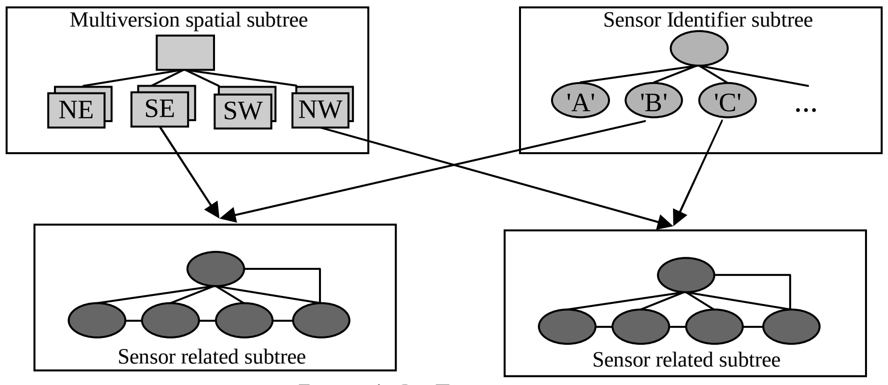

Figure 6: PasTree structure

As shown in figure 6, the PasTree is to use different subtrees to index the data. As shown with the PoTree, storing the data according to the sensors can help in saving processing time, both on updates and other queries. However, strict separation of temporal and spatial substructures is not welcome with agile sensors, sensors that can change of position through time.

Furthermore, while the PoTree provides new spatio-temporal querying patterns, some queries still require the use of sensor identifiers. Just as the MV3R [16] enabled to query the structure according to different perspectives, the PasTree shall have to provide different querying patterns. A secondary structure could have been developed for the PoTree to access the temporal subtrees directly through the sensor identifiers. Such a structure has been integrated into the PasTree design, labeled Sensor Identifier subtree in figure 6.

The spatial subtree is to be changed so as to take into account multiversion properties [17]. While the kd-tree offers interesting properties for fixed structures, the PasTree shall use multiversion quadtrees [18] to index the sensor position through time. This substructure, labeled Multiversion spatial subtree in figure 6 shall integrate some temporal properties.

The sensor related subtree (sensor subtree in figure 6), copied from the temporal subtree of the PoTree has to be improved so as to include two distinct kinds of data. Most updates shall be made from fixed positions, the sensors seldom changing positions. Therefore, it shall not be necessary to keep track of the positions of spatially fixed updates. On the other hand, the sensors may change of position between two updates. This position change is sent to the PasTree structure. The sensor subtree shall have to keep track of the spatial changes so as to provide the possibility to follow the sensor movements through time thanks to its sensor identifier.

Adding an abstraction layer on this subtree could do this. Normal updates shall be considered as simple data while movements shall be recorded as a temporal timestamp, a new spatial position, and a link to the previous known spatial position. Furthermore, on the leaf node level, a link shall point to the last position record of the sensor for this node temporal interval.

These subtrees shall induce data duplication when the sensors shall change of position, with the addition of temporal components to the spatial subtree and of spatial component to the temporal subtree. However, this shall also allow for more detailed queries, based both on spatio-temporal properties and on sensor identifiers.

However, new querying patterns shall emerge, taking advantage of the spatio-temporal queries and of other more classical queries. The PasTree aims at answering queries such as “Follow the different position of the sensor X47 between the time T1 and T2”, “What were the movements between T1 and T2 of the 2 closest sensors from the point <X1,Y1> at time T1” or “What are the data of the sensors in the region defined by the rectangle <X1,Y1 ;X2,Y2> in the last 10 seconds.”

While the development of the PasTree shall lead to new tools for monitoring natural phenomenon, other developments should also be studied to improve the global structure.

FUTURE DEVELOPMENTS

While our work focuses on indexing technologies, other factors must be taken into account in order to address Natural Risk Monitoring through databases.

- Sensor failures should be detected rapidly.

- Data quality should also be analyzed, according to the quality level required, the drift in sensor measurements and their frequency.

- Data localization must be taken into account. Different studies survey the differences between centralized and distributed databases.

- Database saturation is finally a major concern for high frequency updates. Solutions must be provided to store data without clogging up the database.

CONCLUSION

To conclude, our work focuses on Natural Risk Monitoring. We propose a database indexing solution, the PoTree that allows fast spatio-temporal updates in real-time and also real-time spatio-temporal queries. It does so using different subtrees to discern the spatial and temporal (sensor related) dimensions and focusing on the newest data. The PasTree (evolved PoTree) is to add new capacities to the structure by redefining the subtrees. Future works shall focus on database saturation, amongst other things.

REFERENCES

1. Bentley J.L., Multidimensional binary search trees in database application, IEEE Transaction on software engineering, 5(4), 1975, pp. 333-340

2. CENAPRED, Centro Nacional de Prevencion de Desastres [online], http://tornado.cenapred.unam.mx/mvolcan.html , last seen on 04/15/2005

3. Eiman, E., Research Directions in Sensor Data Streams: Solutions and Challenges, technical report DCIS-TR-527, RutgersUniversity, 2003

4. Guttman A., 1984, R-trees: a dynamic index structure for spatial searching. In Proc. 1984 ACM SIGMOD International Conference on Management of Data, Boston, 1984, pp. 47-57

5. Hadjieleftheriou, M. Spatial Index Library [online], http://www.cs.ucr.edu/~marioh/spatialindex/, last seen on 04/15/2005

6. Intanagonwiwat, C. et al., Impact of Network Density on Data Aggregation in Wireless Sensor Networks, in Proc. International Conference on Distributed Computing Systems, Vienna, 2001

7. K-NET, Kyoshin Network [online], http://www.k-net.bosai.go.jp/k-net/index_en.shtml, last seen 04/15/2005

8. Lam, K.Y.and Kuo, T.W., Real-Time Database Systems: Architecture and Techniques, Kluwer Academic publishers, London, ISBN 0-7923-7218-2, 2001

9. Mokbel M. et al., Spatio-temporal Access Methods, IEEE Data Engineering Bulletin, 26(2), 2003, pp. 40-49

10. Mockbel M. F. et al., PLACE: a Query Processor for Handling Real-Time Spatio-Temporal Data Streams, in the VLDB Journal, 2004, 2004

11. Noël, G. et al, The Po-tree, a Real-Time Spatiotemporal Data Indexing Structure, in Proc. of development in Spatial Data Handling SDH2004, Leicester, August, 2004, pp. 259-270

12. Pelekis, N. et al., Literature review of Spatio-Temporal Database Models, in Knowledge

Engineering Review (to appear), 2005

13. Ramamrithan, K. et al., Real-time Databases and Data Services, Journal of Real-Time Systems, vol 28, 2004, pp. 179-215

14. Ratnasamy, S. et al., Data-Centric Storage in Sensornets, in Proc. of the First ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, 2002

15. Satnam, A. et al., A Methodology for Intelligent Sensor Measurement, Validation, Fusion, and Fault Detection for Equipment Monitoring and Diagnostics, Journal of Artificial Intelligence for Engineering Design, Analysis and Manufacturing, vol 15, n°4, 2001, pp. 307-320

16. Tao, Y. and Papadias, D., MV3R-tree: A Spatiotemporal Access Method for Timestamp and Interval Queries. In the VLDB Journal , 2001, pp. 431-440

17. Tao Y. et al., Cost Models for Overlapping and Multi-Version Structures. in ACM Transactions on Database Systems (TODS), Vol 27, n°3, 2002, pp 299-342

18. Tzouramanis, T. et al., Multiversion linear quadtree for spatio-temporal data. In Proc. of ADBIS, 2000, pp. 279-292

19. USGS, U.S. Geological Survey Volcano Hazards Program [online], http://volcanoes.usgs.gov/, last seen 04/15/2005

20. Vega Lopez, I. F. et al., Spatiotemporal Aggregate Computation : a Survey, in IEEE Transactions on Knowledge and Data Engineering, Vol 17, n°2, 2005, pp. 271-286

AUTHORS INFORMATION

Guillaume NOEL	**Sylvie SERVIGNE**	**Robert LAURINI**
noel.guillaume@insa-lyon.fr	sylvie.servigne@insa-lyon.fr	Robert.laurini@insa-lyon.fr
LIRIS, INSA Lyon	LIRIS, INSA Lyon	LIRIS, INSA Lyon