

# Spatial Distributions Server Based on Linear Quadtree

Piotr Bajerski

Computer Science Institute, Silesian Technical University  
16 Akademicka St., 44-101 Gliwice, POLAND  
bajerski@zti.iinf.polsl.gliwice.pl

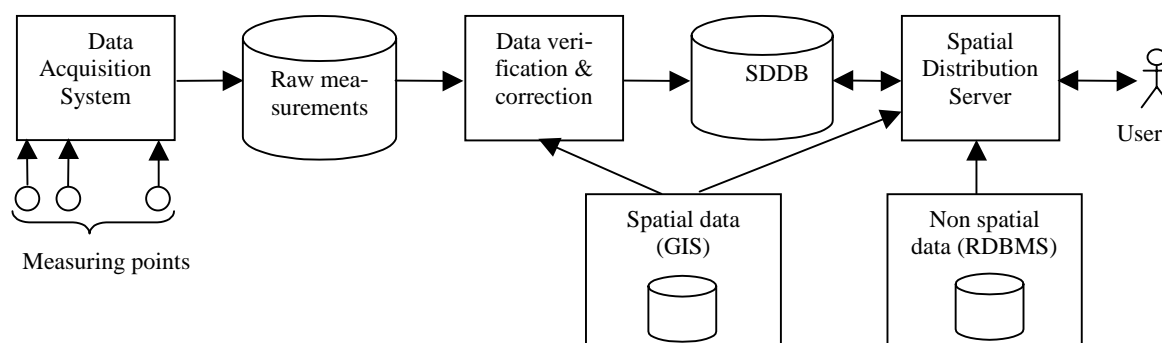
**Abstract.** The paper presents Spatial Distributions Server – an analytical tool aimed at efficient creation, storing and analyzing of spatial distributions. Methods of approximation of spatial distributions by means of quadtree ordered by Peano N curve are presented. Query optimization rules specific to spatial distributions are discussed.

## 1. Introduction

The term *spatial distribution* denotes a set of geographic observations representing the values or behavior of a particular phenomenon or characteristic across many locations on the surface of the earth [6] (e.g. distribution of CO<sub>2</sub> over Upper Silesia). Spatial distribution is a specific kind of spatial data consisting of a set of areal objects with complex borders and many holes. These objects are usually concave. A spatial distribution that exists in reality will be called a real distribution and its computer model used in query processing will be termed an approximation of a distribution or just a distribution. A distribution can be described using vector, tessellated, hybrid or analytical representations [5]. The most common examples of tessellating representations are regular tessellations (grid or raster) and nested regular tessellations (region quadtree). The latter is used in the presented approach. In the research some assumptions about distributions processed were made:

1. only two-dimensional (2D) distributions are taken into account,
2. an approximation of an unknown real distribution is computed using linear local interpolation methods [7, 10, 11] and numerical values obtained at measuring points,
3. queries concern disjoint intervals of distribution values,
4. computations are carried out in a discrete space with the resolution given by a user.

Assumption 3 splits the analyzed area into disjoint regions related to different intervals. The assumption 2 demands that the real distribution is continuous and doesn't change rapidly between measuring points. If a distribution model uses other parameters than values measured (e.g. atmospheric condition) or linear local interpolation methods are not appropriate, then values at additional points must be computed. Also, if values measured are not accurate than they should be corrected using specialised tools before a distribution creation.



**Fig. 1.** Work environment of Spatial Distribution Server

The program aimed at efficient creation, storing and analyzing of spatial distributions is called Spatial Distributions Server (SDS). Fig. 1 presents a typical work environment of SDS. Data gathered by Data Acquisition System, termed raw measurements, are stored in a database. Next the values measured are verified, corrected and supplemented so the assumption 2 holds. During this step a phenomenon model may use data from GIS. Verified data sets are stored in spatial distribution database (SDDB). SDDB keeps also distribution metadata, precomputed distributions and approximations of other spatial objects.

A user communicates with SDS by means of SD SQL – SQL extended with constructions for distributions processing. SD SQL allows writing queries as well as creating and dropping distributions. The SQL *Select* command is extended to express topological and geometric conditions on distributions and other spatial objects. The

fact that a query may include distribution creation has a significant impact on query optimization. A typical query using non-spatial attributes from relational database and data from GIS in vector format is shown below:

*Show Upper Silesia districts with more than 50000 inhabitants where average yearly concentrations in ambient air of suspended matter exceeds norm four times and of Pb exceeds norm two times.*

Attributes of a spatial distribution can be divided into two groups: describing areas belonging to intervals and describing distribution creation process (the measure data set, the generation method and its parameters, the end points of intervals, the coordinates ranges and the resolution). From the user point of view the term distribution denotes the first group of data. The second group is treated as metadata and is stored in a special dictionary of distributions.

A quadtree [12] ordered by a space-filling curve is called *linear quadtree*. Under the assumption that the considered area is split up into homogenous squares, the Peano N curve was chosen to order the quadrants due to simple mapping between coordinates and Peano key – the key is created by bit interleaving of co-ordinates. An approximation ordered by Peano N curve can be saved in a database in a Peano relation so each tuple represents a quadrant [8, 9] or in compact form in such a way that one tuple stores a group of quadrants [3]. The Peano N curve is also called Z-order and Morton ordering.

*Peano-tuple algebra* offers some tools to manipulate Peano relations [9]. Boolean operations (union, intersection and difference) should be available in SD SQL. Geometric operations (translation, rotation, scaling, symmetry, window extraction, replication and simplification) as well as Peano join are used during query optimization and execution.

## 2. Spatial distribution approximation by means of linear quadtree

The analyzed area is divided into elementary quadrants identified by Peano key. Not elementary quadrants correspond to continuous ranges of Peano key. An approximation of a spatial distribution can be interpreted as a function (denoted by *sd*) which assigns to every proper range of Peano key an interval identifier or the value *-1*, indicating that a quadrant has not been classified and it should be split. A range of Peano key is proper if it corresponds to a quadrant yielded by recursive decomposition.

Two approaches to approximation computing were developed [4]:

- bottom-up – first a raster approximation is computed (a function *sd* is iteratively applied to each elementary quadrant) and next it is merged into a quadtree,
- top-down – first whole area is represented by one quadrant and next a function *sd* is recursively applied to each quadrant to check whether it can be assigned to an interval or should be split; the algorithm ends when all quadrants are assigned to intervals.

A function *sd* should minimize computations and error measured as the number of wrongly classified elementary quadrants in respect to a raster approximation. All classification methods developed test if the given quadrant contains measuring points with values belonging to different intervals. Next they use one of the two approaches:

1. check values at measuring points in quadrant neighbourhood and values computed at special points,
2. estimate distribution changes – a quadrant is assigned to an interval if the value interpolated at the quadrant center increased and decreased by an estimate of the distribution change on the distance equal to the half of the quadrant diagonal belongs to the same interval. The distribution change is estimated using an experimental variogram or a model based on normal distribution.

When an elementary quadrant contains values belonging to different intervals, it can be classified basing on average, minimal or maximal value according to the user preferences.

The methods differ significantly in execution time, accuracy and the number of quadrants in distributions approximations. Bottom-up approach is more accurate but is more time and memory consuming. Top-down approach allows using special optimization techniques during query execution. Generally, methods using estimates cause many useless splits and final merging is needed. The best compromise between speed and accuracy offers the method taking into consideration values measured and values interpolated at the centers of the four smaller quadrants. Which method is to be used should depend on whether the user prefers accuracy, quick response or a compromise between these two.

## 3. Spatial Distribution Server Architecture

Fig. 2 shows SDS Architecture. User interacts with a specialised *Graphical User Interface*, called *SD GUI*. It consists of two modules: *Domain Query Wizard* and *Query Formater*. The former module facilitates interactive query formulation using domain terminology and translates it into SD SQL; the latter module formats the query

results and presents them as a map or a table. At SDS the query is first converted into a parse tree. Preprocessor checks if names used in the query are valid and flags them depending whether they are related to spatial distributions, non-spatial attributes or spatial data managed by GIS. Preprocessor yields a valid parse tree or returns an error. Next the query is translated into algebraic expression using relational algebra and Peano algebra operators. Query optimization is covered in chapter 4. Execution Engine performs operations concerning spatial distributions and interacts with other servers according to chosen physical plan. The only module depending on GIS language is *GIS Language Code Generator* as SD SQL uses its own notation for topological and geometric operations.

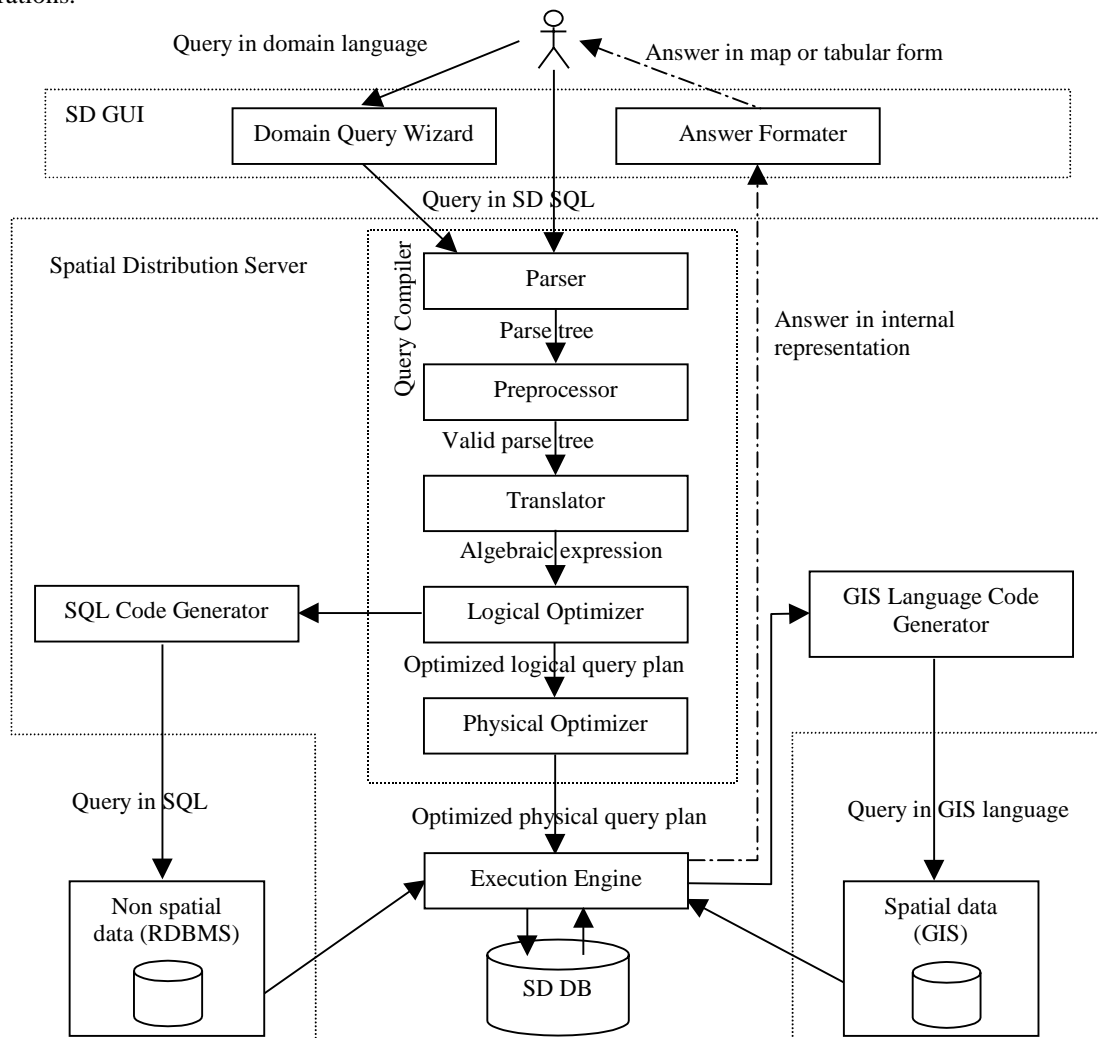


Fig. 2. Spatial Distribution Server architecture

#### 4. Query optimization

Query optimization employs strategies emerging from the nature of spatial distributions [2] and rules adopted from relational databases and from GIS systems. This chapter outlines optimization strategies specific to spatial distributions. The optimization process is divided into two stages: logical optimization when algebraic transformations are applied and physical optimization when the best versions of operators are chosen. At both stages statistics, such as the number of quadrants belonging to intervals and Peano key ranges for clusters of quadrants, are used. The optimiser must take into account two goals: execution time and exactness. The problem of exactness comes up, for example, when the optimiser can choose between transformation of a distribution stored in the database or generation of the distribution in the coordinates and the resolution given in the query.

During logical optimization a query is decomposed according to the observation that the cost of a single operation on an object is lowest for non-spatial attributes, for the vector representation operations are time consuming but more accurate than in tessellated representation, and for tessellated representations data retrieval dominates. Subqueries related to non-spatial attributes are sorted out and sent to relational server. Their results

are used to eliminate some spatial objects. As for operations on vector data, there are two possibilities: execute them in vector format in GIS system or read the data, approximate it and then execute operations. Very important is combining spatial distribution creation with query processing. It allows computing only those parts of distributions that are needed to answer the query. There are at least two cases when this can be done:

- when query concerns only selected intervals – a quadrant can be discarded as soon as it can be stated that it belongs to not selected intervals,
- when an intersection of more than two distributions is computed – a quadrant can be discarded if it does not belong to the intersection already found.

When a query contains conditions on adjacent intervals of a distribution they are treated as one interval. It especially influences on-line distribution generation as a broader interval demands less computation. Because computations are performed in a discrete space de Morgan's laws can be used to replace an intersection of the chosen intervals of some distributions by sum and difference operations. This transformation is cost-effective when the output of the intersection is relatively large.

During physical optimization it must be decided: if objects approximations should be merged, in what order distributions should be taken during an intersection computation, whether the intersection should be performed by SDS or written as SQL statement and performed as Peano join on SD DB [1], and whether a distribution should be read from SD DB or created. During this stage the optimal version for each logical operation is chosen basing on estimates and a conformance level [9] demanded by the subsequent operation.

## 5. Conclusions

Experiments were carried out using data about ambient air pollution over Upper Silesia. The results showed that queries run against distribution approximated by a linear quadtree are executed faster than in vector representation [3]. The most significant improvement was for computing intersections of objects (more than 100-times faster). It is attributed to the replacement of computational geometry by testing relations between line segments on the Peano curve. Data retrieval for both representations takes nearly the same time. Using minimal bounding rectangles for spatial distribution in the vector representation does not give significant improvement if large part of the area belongs to chosen intervals (which very often happens).

The usage of a quadtree for a spatial distribution approximation was compared, in terms of speed and exactness, against raster and vector (isolines) approximations computed using kriging interpolation [7] and local inverse distanced weighted mean interpolation [10, 11]. The error was calculated as the number of pixels assigned to wrong intervals. Broadly speaking the proposed methods are faster than computation on raster giving 3-6% wrongly classified pixels and as quick as isolines computation with slightly bigger error [4].

## Acknowledgement

Part of the research was sponsored by KBN Research Grant 8T11C 028 12.

## References

- [1] Bajerski P.: On using SQL query language for writing spatial queries based on objects approximation (in polish), Silesian Technical University Papers in Computer Science Volume 37, Gliwice 1999.
- [2] Bajerski P.: On optimization of spatial queries based on objects approximation (in polish), Silesian Technical University Papers in Computer Science (in printing).
- [3] Bajerski P.: Efficiency comparison of spatial join of area objects in vector and tessellated representation (in polish), Silesian Technical University Papers in Computer Science (in printing).
- [4] Bajerski P.: Report from KBN Research Grant 8T11C 028 12. Using quadtrees and Peano-tuple algebra for air pollution distribution presentation and processing (in polish), Silesian Technical University, Gliwice 1998.
- [5] Breuning M.: Integration of Spatial Information for Geo-Information Systems. Springer 1996.
- [6] Goodall B.: Dictionary of Human Geography. Penguin 1987.
- [7] Isaaks E.H., Srivastava R.M.: An Introduction to Applied Geostatistics, Oxford University Press 1989.
- [8] Laurini R., Françoise M.: Spatial database queries: relational algebra versus computational geometry. Proceedings of the Fourth International Conference on Statistical and Scientific Database Management, Rome, Italy 1988, M. Rafamelli et al.,(eds) Berlin;Germeny: Springer Verlag. pp. 291-313.
- [9] Laurini R., Thompson D.: Understanding GIS, Academic Press Limited, third printing 1994.
- [10] Nielson M.: Scattered Data Modelling, "IEEE Computer Graphics & Applications", Vol. 1, 1993.
- [11] Sabin M.: Contouring – the State of the Art, "Fundamental Algorithms for Computer Graphics", Springer-Verlang Berlin 1985.
- [12] Sammet H.: The Design and Analysis of Spatial Data Structures. Addison-Wesley, Reding 1989.