

Spatial Data Warehouse and Mining

Rajiv Gandhi

Roll Number 05331002
Centre of Studies in Resource Engineering
Indian Institute of Technology Bombay
Powai, Mumbai -400076 India.



As part of the first stage presentation of M. Tech Thesis
Project
Under the guidance of **Dr. (Mrs.) P. Vankatachalam**

Index

<i>Topic</i>	<i>Page</i>
1. Report Summary	3
2. Introduction	4
3. Motivation	5
4. Challenges	7
5. Related Work	7
5.1 Map cube	7
5.2 Selective Materialization	10
5.3 Spatial Indexing With Pre-Aggregated Results	12
5.4 Work Related to Spatial Data Mining	13
6. Work Done	14
6.1 Literature Survey	14
6.2 Proposed Approach for OLAP	14
6.3 Proposed Approach for Association rule	15
7. Works to be done	16
8. References	17

1. Report Summary

Due to enormous volume of spatial data, it becomes difficult to analysis that volume without having specially designed softwares. Recently, the popularity of spatial information, such as maps created from satellite images and the utilization of telemetry systems, has created repositories of huge amounts of data which need to be efficiently analyzed. Spatial data warehouse is answer to this problem

In analogy to the non-spatial case, a spatial data warehouse can be considered, which supports On Line Analytical Processing (OLAP) operations on both spatial and non-spatial data; this is different from classical data warehouse because data domain is changed.

Various OLAP operations are done on spatial data for decision making and only efficient spatial data warehouse can provide better OLAP operations. However, having an efficient warehouse design itself poses many challenges because of data format.

On the other hand, discovery of spatial association rules may disclose interesting relationships among spatial and/or nonspatial data in large spatial databases and thus it represents a new and promising direction in spatial data mining. Spatial data mining techniques are often derived from spatial statistics and spatial analysis. Other integrals of mining tools are machine learning and database, which are customized to analyze massive data set.

In this project, I aim to this problem by implementing suitable method from the field of Spatial Data Warehouse and Mining. This report gives a description of the preliminary work done in this direction.

Keywords: Map Cube, Data Cube, Spatial Data Mining, Spatial Data Warehouse, OLAP, GIS, Knowledge discovery

2. Introduction

Today, we have terabytes of data processed in database systems and we store a measurable portion of that data for analysis purpose and this trend is increasing as new technologies are arriving [K. Koperski, J. Han 1995], for example, wide applications of remote sensing technology and automatic data collection tools store data in large spatial databases.

Existing data organization and retrieval tools can only handle the storage and retrieval of explicitly stored data. So there is a constant desire to have such software tools which can handle huge amount of data and make analysis and decision making easier. Data warehouse and data mining techniques came up and above said problem is being solved day by day more efficiently through research. Data warehouses are collections of historical, summarized and non-volatile data, which are accumulated from transactional databases. (Transactional database is collection of records and specially designed to handle large volume of transactions concurrently)

But again, requirements of mining of spatial database are different from those of mining classical relational database, in particular, the notation of spatial autocorrelation where similar objects tend to cluster in geographic space, is central to spatial data mining [S. Shekhar].

A spatial data warehouse can be considered, repository of multiple heterogeneous data sources, organized under a unified schema at a single site in order to facilitate management decision making [Han Kamber], which supports OLAP (On Line Analytical Processing) operations on spatial and non-spatial data (i.e. slice, dice, drill down, roll up etc.), as well as make easier for algorithms to find hidden patterns. Although OLAP tools support multidimensional analysis and decision making, additional data analysis tools are required for in-depth analysis [S. Shekhar et al. 1999].

The reminder of this report is organized as follows: section 3 gives the motivation for working in this area. Section 4 formally describes the problems while making Spatial Data Warehouse and Mining, highlighting some of the challenges that we are facing followed by a brief survey of the related work done in this area, in section 5. Section 6 gives some of the possible approaches that can be considered a good choice while working on the said domain. Finally section 7 lists out the future plan of entire project.

3. Motivation

While working with GIS softwares tools, a user wants to take decision based on the analysis of huge data, but due to human limitation it is expected that the software should do much of the work. So, it is desired to perform analysis and decision making without much user intervention. Following are some of the applications of spatial data warehouse which motivate us to work on this domain. Although these examples can be implemented on GIS software but the response time is main critical factor which is being considered in our study.

3.1 Overlay of Multiple Thematic Maps [Nebojsa, Jaiwei, Krzysztof 2000]

We often have multiple thematic maps in a spatial database, like altitude map, population map, and daily temperature maps of a given region. By overlaying these multiple thematic maps or in other words making the map cube of thematic maps, one may want to find some interesting relationships among altitude, population density, and temperature by doing OLAP operations. User may like to perform analysis on any selected dimension, such as drill down along a region to find the relationships between altitude and temperature.

3.2 Traffic System [D. Papadias et al. 2001]

Consider a traffic supervision system that watches the positions of cars in a city and the road traffic. A car can be either on a road segment or it can be in a parking lot. The goal of such a system is: "find the road segments with the least traffic near the center" or, given a medical emergency, "which is the hospital that can be reached faster given the current traffic situation". In both cases, it is statistical information, i.e., the number of cars, rather than their ids that is important. Furthermore, the extraction of this information can be time consuming. Consider that the positions of the cars are stored in an R-tree R_C , and the extents of the line segments in a tree R_R .

Answering a query such as "give me the traffic for every road segment in an area of 1km radius around each hospital" would require a spatial join between R_C and R_R . The same system could also be used to answer queries

involving fire emergencies, in which case the areas of interest would be around fire departments, police stations and so on.

3.3 Regional Weather Pattern Analysis [Nebojsa, Jaiwei, Krzysztof 2000]

In most of the countries, they have many weather probes scattered, each recording daily temperature and precipitation for a designated small area and transmitting signals to a provincial weather station. A user may like to view weather patterns on a map by month, by region, and by different combinations of temperature and precipitation, or even may like to dynamically drill-down or roll-up along any dimension to explore desired patterns.

4. Challenges

Section 3 gives some of the examples describing the problems in the domain of spatial data warehouse and mining. Here we describe few more issues to understand the challenges involved in this working domain.

- Construction of spatial data warehouses is done by integration of spatial data from heterogeneous sources and systems. Spatial data is usually stored in different industry firms and government agencies using different data formats. Data formats are not only structure specific (e.g., raster- vs. vector-based spatial data, object oriented vs. relational models, different spatial storage and indexing structures, etc.), but also vendor-specific (e.g., ESRI, MapInfo, Intergraph, etc.) [Jaiwei, Nebojsa, Krzysztof 1998]. Moreover, even with a specific vendor like ESRI, there are different formats like Arc/Info and ArcView (shape) files. There has been a lot of work on data integration and data exchange and it is still going on.
- The realization of fast and flexible online analytical processing in a spatial data warehouse is the main objective of this project. In spatial database research, spatial indexing and accessing methods have been studied extensively for efficient storage and access of spatial data [Nebojsa, Jaiwei, Krzysztof 2000]. Unfortunately, these methods alone cannot provide sufficient support for OLAP of spatial data because spatial OLAP operations usually summarize and characterize a large

set of spatial objects in different dimensions and at different levels of abstraction, which requires fast and flexible presentation of collective, aggregated, or general properties of spatial objects. Like in the case of OLAP of relational data, in order to achieve adequate performance, it is necessary to pre-compute and store some aggregates where as Spatial indexing and accessing methods are not designed for such tasks which in result make the things to be customized according to data formats. So making generic solution is difficult.

- In OLAP we emphasis to reduce the response time which is necessary because of heavy computation while computing results of operations. We can pre-compute selected data which is often used, but we can't pre-compute all data due to volume and constraint of disk space. So which to be pre-computed? A major challenge towards the implementation of OLAP

5. Related work

This section presents some of the work done related to this area, moreover, closely related to our motivating examples.

5.1 Map cube [S. Shekhar et al. 1999]

Map cube plays an important role in the process of OLAP operation. The data in the warehouse are often modeled as a multidimensional space to facilitate the query engine for OLAP. Map cube is an operator which takes a base map, associated data table, aggregation hierarchy and cartographic preferences to produced album of maps. With the map cube operator we visualize the data cube in spatial domain via album of maps.

The basis of map cube is the hierarchy lattice, a dimension power set, concept hierarchy or it can be both. But the main work has been done on dimension power set. Proper grammar has been proposed to create map cube operator which takes care of cartographic preferences, aggregation hierarchy and input maps. Figure 1 describes the whole process for generating Map Cube.

In OLAP operation, map cube operator supports the following operations.

- **Roll up:** this operator generalizes one or more dimension and aggregates the corresponding measures by increasing the level of abstraction.
- **Drill down:** it specializes in one or few dimensions and presents low-level aggregation by decreasing the level of abstraction or increasing the detail.
- **Slice:** when we drill into one level down in one dimension that is called slicing, but the number of entries displayed is limited to that specified in the slice command.
- **Dice:** A dice operation is like slice on more than one dimension, i.e. on a 2-dimension display, dicing means slicing on the row and column dimensions.

In other words, a map cube is a data cube with cartographic visualization of each dimension to generate an album of related maps for a dimension power- set or a concept hierarchy. A map cube adds more capability to traditional GIS where maps are often independent identity.

5.1.1 Mathematical Notation for Dimension Power Set

Let the number of dimensions be m and each dimension be A_i where $i = 1, 2, \dots, m$ and A_m be the geographic location dimension. Then we have $(m-1)$ different level of lattice for the dimension of power set hierarchy. Now let the level with only one dimension A_i where $i = 1, 2, \dots, m-1$ be the first level and level with two dimension is A_{ij} where i not equal to j and $i, j = 1, 2, \dots, m-1$. Now complete set of dimensions $A_1, A_2, A_3, \dots, A_{m-1}$ will be at the $(m-1)^{th}$ level. Then the total number of cuboids are

$$\sum_{i=1}^{m-1} C_i^{m-1}$$

Now let the cardinality of each dimension A_i be C_i then for each cuboids then we have $C_i \times C_j \times \dots \times C_k$ maps

The data cube capability of roll up, drill down, slice and dicing combined with the map view, benefits the analysis and decision making process based on spatial data warehouse.

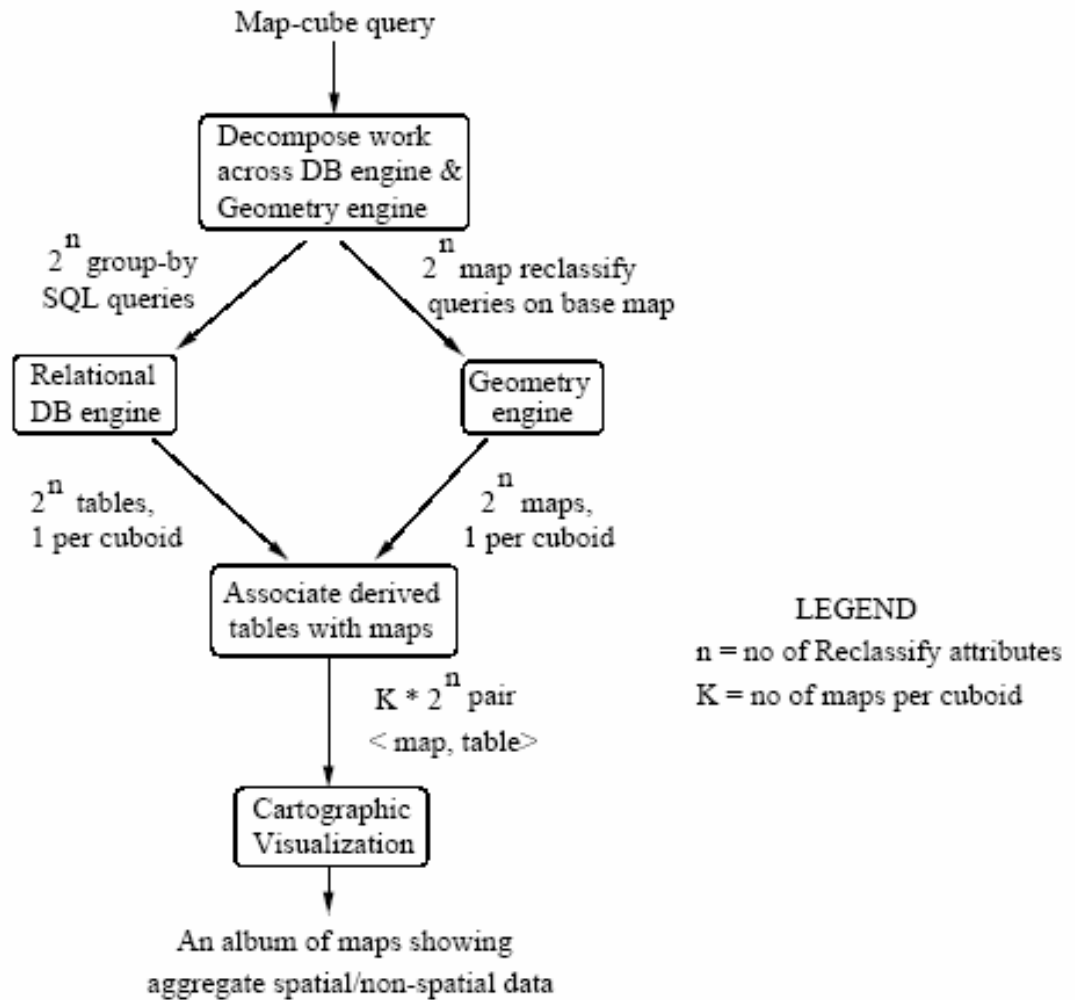


Figure 1: Steps in Generating a Map Cube [S. Shekhar et al. 1999]

5.2 Selective Materialization [Jaiwei, Nebojsa, Krzysztof 1998] [Nebojsa, Jaiwei, Krzysztof 2000]

Since a cuboid usually consists of a large number of spatial objects, it may involve pre-computation and storage of a large number of mergable spatial objects but some of them could be rarely used. Therefore, it is recommended to perform selection at a finer granularity level by examining

each group of mergable spatial objects in a cuboid to determine whether such a merge should be pre-computed. Assumption has been taken that the computation of spatial measures involves spatial region merge operation only, but it is also stated that these are also applicable to other kinds of spatial operations, such as spatial map overlay, spatial join and intersection between lines and regions.

There can be at least three possible choices regarding the computation of spatial measures in spatial data cube construction:

1. Collect and store the corresponding spatial object pointers but do not perform pre-computation of spatial measures in a spatial data cube. This choice indicates that the (region) merge of a group of spatial objects, when necessary, may have to be performed on-the-fly.
2. Pre-compute and store some rough approximation/ estimation of the spatial measures in a spatial data cube. Such a pre-computed result is as small as a non spatial measure and can be presented quickly to users.
3. Selectively pre-compute some spatial measures in a spatial data cube. This seems to be a smart choice. However, the question is how to select a set of spatial measures for pre-computation. The selection can be performed at the cuboid level, i.e., either pre-compute and store each set of mergable spatial regions for each cell of a selected cuboid, or pre-compute none if the cuboid is not selected..

The following heuristics are used for selection of groups of connected regions to be premerged.

- Access frequency.
- Cardinality of a group of connected regions
- Sharing among the cuboids in the cube lattice structure.

Let F and G be groups, containing pointers to spatial objects such that G is subset of F. Then group F is a non-occluded (non-blocked) ancestor of G, only if the following conditions are satisfied:

- Group F has not been materialized
- There is no materialized group J such that $G \subset J \subset F$
- There is no materialized group J, $J \subset F$ s.t. $G \cap J \neq \emptyset$. & cardinality of J > cardinality of G

5.2.1 Algorithm (Spatial Greedy Algorithm).

A greedy algorithm which selects candidate (connected) region groups for premerging in the construction of a spatial data cube.

Input

A cube lattice which consists of a set of selected cuboids (presented as nodes) obtained by running a cuboid selection algorithm, such as HRU Greedy.

- An access frequency table which shows the access frequency of each cuboid in the lattice.
- A group of spatial pointers in each cell of cuboids in the lattice.
- A region map which delineates the neighborhood of the regions. The information is collected in an table
- `max_num_group` as the maximum number of groups which are expected to be selected for premerge.

Outline of Algorithm

- Firstly find the connected groups send `candidate_table` as a input parameter
- Intilize `merged_obj_table = ∅`
- `Remaining_set = candidate_table`
- REPEAT
- `Select_candidate (remaining_set, merged_obj_table)`
- UNTIL `cardinality (merged_obj_table) >= max_num_group`

5.3 Spatial Indexing With Pre-Aggregated Results [D. Papadias et al. 2001]

The aggregation functions are divided into three classes: *distributive*, *algebraic* and *holistic*. Spatial index is built on the objects of the finer granularity in the spatial dimension and the groupings of the index are used to define a hierarchy. This implicit hierarchy is built in the lattice model to select the appropriate aggregations for materialization. A study of several

algorithms for spatial aggregation have been referred and proposed a method which traverses the index in a breadth-first manner in order to compute efficiently group-by queries [R. Agrawal, R. Srikant 1994] and employed incremental update techniques.

5.3.1 The Aggregation aR-Tree [D. Papadias et al. 2001] Structure

The aggregation aR-Tree, stores for each minimum bounding rectangle (MBR), the value of the aggregation function for all the objects that are enclosed by the MBR. The R-tree is built on the finest granularity objects of the spatial dimension. An important property of the R-tree is that every object at level $l-1$, belongs to exactly one MBR at level l . Figure 2 depicts an aR-tree which indexes a set of five road segments, $r_1 \dots r_5$, whose MBRs are $a_1 \dots a_5$ respectively. There are three cars on road r_2 , therefore there is an entry $(a_2, 3)$ in the leaf node of the aR-tree. Moving one level up, MBR A_1 contains three roads, r_2 , r_3 and r_5 . The total number of cars in these roads is six; therefore there is an entry $(A_1, 6)$ at level one of the aR-tree. The general concept can be applied to different types of queries; for instance, instead of keeping aggregated results of joins the aR-tree could store such results for window queries. Furthermore we could employ the same idea to other data partitioning or space partitioning data structures (e.g., Quadtrees).

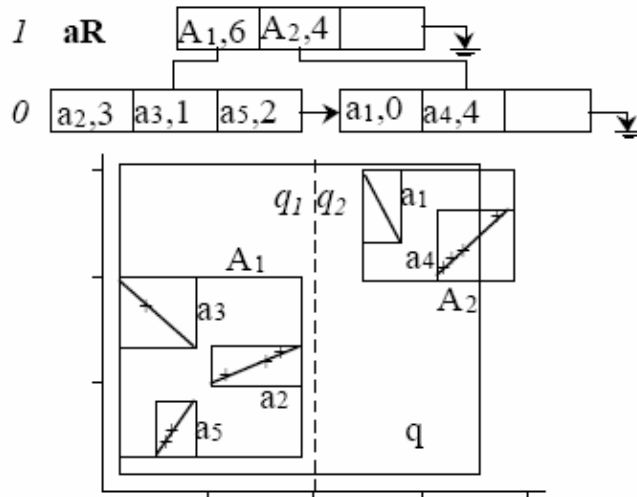


Figure 2: A R-Tree [D. Papadias et al. 2001]

It is straightforward to extend the above definition to handle multiple aggregate functions. Instead of storing one result in each entry of the tree, we store a list of results for all the necessary functions. In our example, if the maximum number of cars in each road is also required, the entry for AI will be $(AI, 6, 3)$ since there are 3 cars on road $r2$. In this way, the aR-tree can also handle algebraic aggregate functions. If, for instance, we need the average number of cars in each road segment covered by a node, in addition to the total number of cars we need to store the number of road segments covered by the node.

5.4 Work Related to Spatial Data Mining

5.4.1 Statistical Analysis

Until now statistical spatial analysis has been one of the most common techniques for analyzing spatial data and has a strong possibility of getting models of spatial phenomena. However, statistical analysis usually requires the assumptions regarding to statistical independence of spatially distributed data [K. Koperski, J. Han 1995]. Such assumptions are often unrealistic due to the influence of neighboring regions but solution might be that the analyst may fit a regression model with spatial lagged forms of the dependent variables [S. Shekhar].

But often statistical analysis also deals poorly with symbolic data and nonlinear rules cannot be described using standard methods in statistical spatial analysis. Statistical approach requires a lot of domain and statistical knowledge [K. Koperski, J. Han 1995]. Another problem related to statistical spatial analysis is expensive computation of the results. Thus, it should be performed by domain experts with the experience in statistics.

5.4.2 Generalization Based Spatial Association Rule

In spatial Data Mining, one major approach is to apply generalization techniques on spatial and non-spatial data to generalize detailed spatial data at certain high levels for studying the general characteristics and data distributions at this level, spatial objects are expressed as merged spatial regions or clustered spatial points. However, these methods cannot discover rules reflecting the structure of spatial objects and spatial-spatial or spatial-nonspatial relationships which contain spatial predicates, such as adjacent to, near by, inside, close to, intersecting, etc. As a complementary, spatial

association rules represent object/predicate relationships containing spatial predicates. For example, the following rules are spatial association rules.

Example 1: Nonspatial Consequent With Spatial Antecedent(S).

$$\text{is_a}(x, \text{car}) \wedge \text{make}(x, \text{Mercedes}) \rightarrow \text{is_expensive}(x): (95\%)$$

Example 2: Spatial Consequent with Non-Spatial/Spatial Antecedent(S).

$$\text{is_a}(x, \text{Flyover}) \wedge \text{size}(x, \text{above } 100 \text{ mtr}) \rightarrow \text{is_made_on}(x, \text{highway}): (85\%)$$

Various kinds of spatial predicates can be involved in spatial association rules. They may represent topological relationships between spatial objects, such as disjoint, intersects, inside/outside, adjacent to, covers/covered by, equal, etc. They may also represent spatial orientation or ordering, such as left, right, north, east, etc., or contain some distance information, such as close to, far away, etc.

6. Work done

6.1 Literature Survey

The literature survey carried out to understand the problem and work that has already been done in this area, have been dealt with in the earlier section. Next, we suggest some possible approaches, for the structure of Map Cube, which efficiently supports OLAP operation such that its response time is reduced as well as finds Association Mining Rules quickly.

6.2 Proposed Approach for OLAP

We need to be able to do indexing, when we access data from selective materialized map cube [Jaiwei, Nebojsa, Krzysztof 1998] [Nebojsa, Jaiwei, Krzysztof 2000], data should be available and filtered on one shot. For this we propose Bitmap Indexing (BI) for stored data. A call to BI will be made after making request to OLAP engine, although BI will be implemented as an internal part of engine, to get relevant data.

6.3 Proposed Approach for Association rule

Previous study [K. Koperski, J. Han 1995] is based on the assumption that a user has prior knowledge on concept or operation hierarchies for nonspatial or spatial generalization. Such assumptions, though valid in many cases, may enforce some strong restrictions to naive users or to some complex spatial databases with poorly understood structures or knowledge. Because of this more studies are needed to overcome these restrictions.

Another suggestion is that there are many algorithms for finding association rules but none of them use Map cube to find association rules, if we try to find association rules in cuboids then situation could be different and we may come across new hidden patterns. Proposed suggestion is based on the following statement, To find Association rules we count the probability of every combination of records, to measure the confidence and support, if we increase the number of records by aggregation in Map Cube then it may turn out that we encounter some more hidden patterns.

7. Works To Be Done

The main objective of this project is to make a prototype for data warehouse and OLAP operations with viewer. The following subsections describe in detail.

- Transformation from Vector Structure to RDBMS

The Vector Structure data, which is already available in Gram++, will be transformed into any RDBMS (MySQL, PostGRES, etc), so that we can eliminate the language (VC++) dependency, which gives freedom to work further with any programming language.

- Making of Map Cube

The spatial data will be organized in R-tree structure to make map cube [S. Shekhar]

- Viewer for Map Cube

A viewer is to be made, possibly in JAVA language, which will show results, after doing OLAP operations.

- Implementation Of OLAP Operation

The prototype will support most of the proposed OLAP operations [D. Papadias et al. 2001] and allow user to see the results in viewer.

- Association Rules

In the data mining part, attempt will be made to develop a new algorithm which finds associations rules, based on previous study [K. Koperski, J. Han 1995], which frees the user from the restriction of having concept hierarchy ready before hand while using Map Cube.

8. References

- [1] Rakesh Agrawal, Ramakrishnan Srikant “**Fast Algorithms for Mining Association Rules**” (1994), Proc. 20th Int. Conf. Very Large Data Bases, VLDB.
- [2] Shashi Shekhar, **E- Book (Draft for 7th chapter)** on the web address of <http://www.cs.umn.edu/research/shashi-group/Book/>.
- [3] Jaiwei Han, Nebojsa Stefanovic and Krzysztof koerski, “**Selective Materialization: An Efficient Method for Spatial Data Cube Construction**” (1998) Research and Development in Knowledge Discovery and Data Mining, Second Pacific-Asia Conference, PAKDD'98
- [4] Nebojsa Stefanovic, Jaiwei Han and Krzysztof koerski, “**Object-Based Selective Materialization for Efficient Implementation of Spatial data cubes**”, IEEE Transactions on Knowledge and Data Engineering , Vol 12, No.6 (Nov/Dec 2000).

- [5] S. Shekhar, C.T. Lu, X. Tan, S. Chawla, R. R. Vatsavai, “**Map Cube: A Visualization Tool for Spatial Data Warehouse**”, (1999)

- [6] Dimitris Papadias, Panos Kalnis, Jun Zhang and Yufei Tao, “**Efficient OLAP Operation in Spatial data Warehouse**” (2001)

- [7] Jiawei Han, Micheline Kamber, Book Title “**Data Mining Concept and Techniques**”, Morgan Kaufmann (An Imprint of ELSEVIER) Publication, ISBN: 1-55860-489-8, 2003

- [8] Krzysztof Koperski, Jiawei Han, “**Discovery of Spatial Association Rules in Geographic Information Databases**”, (1995), Proc. 4th Int. Symp. Advances in Spatial Databases, SSD