

# THE APPLICATION OF DATABASE MINING TECHNIQUES TO DATA FUSION IN SPATIAL DATABASES

*Janusz R. Getta, Lusiana and Phillip J. McKerrow*

Department of Computer Science  
University of Wollongong  
Northfields Ave.,  
Wollongong  
NSW 2522  
Australia  
e-mail: {jrg, Lusiana, phillip}@cs.uow.edu.au

## ABSTRACT

This work applies some of the techniques used in mining of association rules in large relational databases to fusion of data from large spatial databases. The specific objective is to show that algorithms developed for the identification of frequent itemsets in the relational databases can be, after slight modifications, successfully used to fuse complex two dimensional objects from large amounts of range data collected with ultrasonic sensors. The paper presents two algorithms that use this approach to construct polylines and polygons from ultrasonic range data.

## 1. INTRODUCTION

Data fusion is commonly defined as a process of extracting useful information from noisy sensor data [5, 6, 7, 14]. We prefer to view data fusion as a process that synthesis of information about complex real world objects from raw and low level information obtained from external sensors. Data fusion techniques provide an important shift in abstraction from information about properties of primitive objects to properties of the objects at the higher levels of abstraction. Different data fusion strategies strongly depend on the format of input data e.g. signals, maps of pixels, or collections of primitive geometric elements. In our approach we shall consider data fusion from a large set of geometrical primitive elements - line sections - into two dimensional complex geometrical objects.

Database mining is the data-driven extraction of information from large databases [4]. Its objective is to identify the patterns, rules or functions that occur

in large amounts of experimental and measured data. Understanding of data, that is learning about implicit information from explicit data is important for the support of making strategic decision and for technical applications. Database mining also provides a shift in abstraction from information about the elementary facts into a collection of rules that govern the occurrences of these facts.

Our prime objective is to develop the algorithms to fuse ultrasonic range data collected by a mobile robot as it navigates through a room into a formal description of an indoor environment using spatial database techniques. The robots with ultrasonic sensors take sets of readings as they move through the environment. The readings are converted into primitive geometric elements - edges and surface segments [8, 9]. The proposed data fusion algorithms will combine the primitive elements obtained in a random order from several robots into complex two dimensional objects and determine the location of such objects.

An important assumption is that a database of primitive geometric elements is so large that it cannot be entirely stored and processed in fast transient memory. As a consequence of this assumption we have to consider a class of data fusion algorithms that promote an efficient management of persistent memory and that minimize the number of data transfers between transient and persistent memory. Efficient data fusion of complex objects from a large database of primitive geometrical elements requires optimal organisation of spatial structures of persistent memory and minimization of passes through the database. This is the reason why we believe that some of the techniques which are presently used in database mining can be adopted to data fusion in large spatial databases.

---

This work was supported by a Small ARC Grant obtained from The University of Wollongong

The problem of mining association rules out of so called "basket data" has its roots in the processing of massive databases that contain data about customer transactions in large supermarkets [1, 2, 3, 12, 13]. An example of an association rule is an implication saying that "98% of the customers that purchase tires and car accessories also get a car service done". Identification of such rules in commercial data is valuable for marketing, analysis of buying patterns, sales programs and strategies. An important part in the identification of association rules is the discovery of "frequent (large) itemsets". A frequent itemset is a set of objects (items) such that the number of its occurrences in the customer transactions exceeds a certain threshold. Our hypothesis is that the techniques used to discover frequent itemsets are similar to those required for fusion of data in large spatial databases. Thus, its solutions can be used to develop efficient data fusion algorithms.

We observe that at a certain level of abstraction the objectives of data fusion and discovery of frequent itemsets are almost the same. In both cases we need to synthesis the structures of complex objects from large sets of elementary (primitive) objects. In data fusion, membership of a primitive element in a complex object is determined by its position in two dimensional space. In database mining, a frequent itemset is a collection of semantically related elementary items. An item belongs to a frequent itemset if it has "is-part-of", "is-component-of" type of relationship, etc. with the other items from such a set.

Discovery of the frequent itemsets is done by analysis of customer transactions. Is it possible to detect such "transactions" in a data fusion case? In order to answer this questions we have to investigate what does the term "transaction" mean for discovery of frequent itemsets? If we assume an ideal case where the customers always purchase all components of one or more complex objects then each transaction precisely represents such objects. Unfortunately, customers sometimes buy only subsets of complex objects. However these cases do not completely destroy the above definition of transaction. In these cases a transaction is a rough approximation of one or more complex objects.

Discovery of frequent itemsets is equivalent to identification of complex objects from their rough approximations. Therefore, in a data fusion case a transaction should always be a rough approximation of complex geometrical objects. How do we detect such "transactions"? The idea is to divide a space of primitive objects into small partitions and to register which object fits within each partition. Then a description of each partition is equivalent to a transaction. The partitions should be constructed in such a way that their

shapes are rough approximations of the fused objects.

One of the problems is identification of the best partitioning strategy. A regular square grid is one of standard space partitioning techniques recommended by the theory and practice of spatial databases [10]. In our case square grid is not the best approximation of line segments. If we assume that a partition should roughly approximate the objects then its shape should be similar to the shape of the objects. For instance if we expect to fuse long straight lines (e.g. walls) then the elements of grid should be similar to long and narrow rectangles.

The next problem is an efficient analysis of the partitions. One of the local aims in discovery of frequent itemsets is elimination of all transactions that are bad approximations of complex objects. It is achieved by the decomposition of entire set of transactions into several smaller subsets which are "the seeds" for complex objects. Transactions which are inconsistent with a decomposition are skipped. An important property is that all transactions are independent. Such an assumption is not valid in the data fusion case. This is why an analysis of partitions cannot be done in exactly the same way. For instance, if two partitions overlap or if they have a common border then it is possible that both of them represent the same complex object. Information about the relative position of two or more partitions can be used to eliminate some of them or to synthesis an object that overlaps several partitions.

The partitions are constructed in one pass through the entire database of primitive geometrical objects. We test the parameters of each object in order to find all of the partitions that it belongs to. Then, each object is replaced with its identifier which is later written into the representations of respective partitions. Efficiency of this step strongly depends on our ability to minimize the number of partitions an object is inserted into.

In the next step the partitions with the largest number of objects are used to synthesis the initial complex objects. The other partitions that represent only parts of complex objects are discarded. This step requires the comparison of partition properties, and its efficiency depends on the number, size and representation of partitions. The complex objects identified in the first pass are written to a new database. This database is used for fusion of the higher level complex objects in the next passes.

Section 2 describes the organisation of the spatial database and integration of the elementary line segments into grid representation. An algorithm for fusion of polylines from grid elements is presented in Section 3. Finally, Section 4 describes fusion of the polygons from

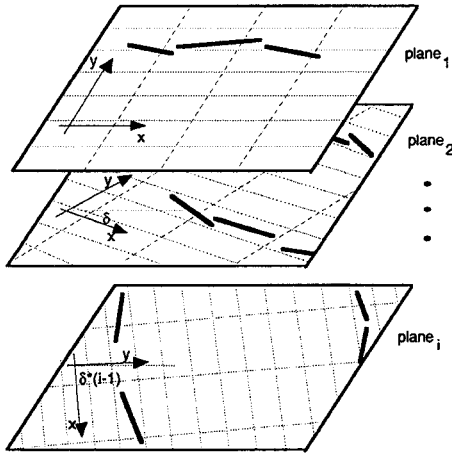


Figure 1: Grid planes

the descriptions of polylines and information about the basic movements of sensing devices.

## 2. DATABASE

Fusion of complex two dimensional objects starts with the integration of elementary line segments obtained from the ultrasonic sensors into a spatial database. The description of elementary line segment is a quadruple  $[p_1, p_2, p_s, p_e]$  where  $p_1$  and  $p_2$  are the endpoints of the segment while  $p_s$  and  $p_e$  are the locations of the sensing device when the current line segment was detected. The direction of line segment is relative to the x-axis of the coordinate system.

A conceptual view of the database is formed by a sequence of  $n$  planes  $p_1, \dots, p_n$  such that the coordinate systems of adjacent planes  $p_i$  and  $p_{i+1}$  are rotated by angle  $\delta$ , where  $n \times \delta = \pi$ , see Figure 1.

Each plane contains information about all of the line segments whose angle lies between the x-axis of plane  $p_i$  and the x-axis of  $p_i + \delta$ .

The planes are partitioned into a uniform rectangular grid whose horizontal lines are parallel to the x-axis and vertical lines are parallel to the y-axis. The size of a single grid element is determined by an assumption that no line segments span over more than two adjacent rows. Therefore, the height of grid element must be greater than  $d \times \sin(\delta/2)$ , where  $d$  is an expected maximum length of segments and  $\delta$  is an angle by which coordinates of plane  $p_{i+1}$  are rotated in respect to coordinates of plane  $p_i$ . The width of grid element is an integer multiplication of its height, that is  $width = k * height$ . All grid elements are divided into  $k$  square zones.

The spatial database of grid elements is implemented in object-oriented database technology. Collection of

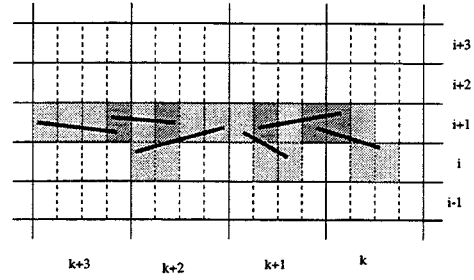


Figure 2: Grid elements and zones

all grid elements is represented in the database by a class of *grid objects*. An instance of grid object consists of logical identifier, the descriptions of all zones in the respective grid element, and descriptions of all places from where the line segments that span over the grid element have been detected.

In order to reduce access time a class of grid objects is partitioned into  $n$  database clusters in such a way that a single cluster contains the instances of all objects representing the grid elements from only one plane. A database cluster is a collection of objects which are physically stored closed to each other, i.e in the same disk block, or in a segment of consecutive disk blocks placed on the same disk track or on the same cylinder.

Further reduction of access time is achieved by indexing the grid objects over their logical identifiers. Such index maps the coordinates of grid elements into the physical addresses of respective grid objects. Logical identifiers of grid objects are constructed by concatenation of string representations of row and column numbers separated by a nondigit delimiter. Each database cluster has a separate index. Separate indexing of clusters decreases size of each index and allows for simultaneous access to more than one cluster. Finally, to speed up a horizontal scan of grid planes, the grid objects representing the grid elements from the same row and adjacent columns are linked by references to their physical addresses.

Object-oriented database management system Ode (v. 4.2) [11] is used for implementation of the database.

Integration of line segments is performed in the following way. For each line segment, the system finds its direction and identifies grid plane  $p_i$  it belongs to. The line segment is rotated by angle  $(-i) \times \delta$  in order to adjust its endpoints to coordinate system of plane  $p_i$ . The endpoints of rotated line segment are transformed into numbers of rows and columns of all grid elements overlapped by a segment. If a segment spans over several grid elements then it is partitioned into a number of smaller segments. Index of cluster representing plane  $p_i$  is accessed to find the physical addresses of respective

grid objects. Approximate position of a line segment is recorded in the database by incrementing all counters associated with all zones overlapped by a segment. At the end of this process the groups of line segments with similar orientations are represented by the grid objects stored in the same cluster. Density of line segments per elementary area of grid element is represented by the respective zone counters. (Figure 2).

Note that zones play a similar role to products purchased by a customer and grid elements are similar to transactions in the problem of "frequent itemsets" mining. Like in the previous case we search for all combinations of zones such that their counters exceed a given threshold level.

### 3. POLYLINES

Fusion of polylines is done in two stages. First, the system constructs the elementary polylines on the separate grid planes. Next, the endpoints of elementary polylines from different grid planes are matched to fuse longer polylines. In the same step, all short polylines whose directions are contradictory to directions of the larger polylines are identified as wrong measurements and are removed from the database.

To fuse elementary polylines the system randomly picks a nonempty grid element on an arbitrarily selected plane. The first step is to check whether the selected grid element actually represents the majority of line sections in this area. In order to identify the grid element that contains the maximum number of line sections we scan vertically through all of the grid planes and compare zone counters of all grid elements in the nearest neighbourhood of initially selected grid element. A grid element with the largest value of all zone counters is adopted as the best starting point.

Next, we move to the next (previous) column on the same grid plane to find the largest rectangular areas of zones with counters greater than a given threshold value. In this process all zones whose counters are less than a given threshold value are discarded. This procedure is repeated for all columns on the left and right hand side of a starting grid element until a column with no adjacent grid elements is found. Rectangles identified in this process are used to construct the elementary polylines in the following way. For any two adjacent rectangles  $r_i$  and  $r_{i+1}$  a point  $p_i(x_i, y_i)$  in the middle of their common border is found. If a rectangle has no left (right) neighbour then the position of the leftmost (rightmost) point is in the middle of its leftmost (rightmost) edge. A sequence of points constructed in such a way determines the corners of an elementary polyline ( see Figure 3 ).

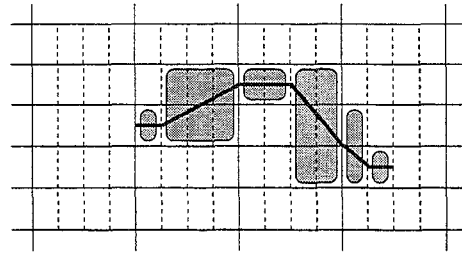


Figure 3: Construction of elementary polyline

It is important to note that different sensors usually provide inconsistent information about position and shape of sensed objects. This is why elementary polylines cannot be treated as exact representation of solid objects. Elementary polyline may be considered as a core of *fuzzy polyline* whose shape and width is determined by the grid elements with nonzero zone counters.

In fact, due to the errors and limited precision of the measurements we encounter the same problem as in "mining" of frequent itemsets. Combinations of items obtained at the end of mining process depend on a threshold value which is used for representation of the term "frequent". In our case, the shape of elementary polylines depends on a threshold value which is used for elimination of zones with the small counter values.

### 4. POLYGONS

The final objective of data fusion is to construct the polygons that represent free and occupied areas of the analyzed space. Fusion of the polygons is based on information about polylines, sensing routes and sensing directions obtained from the analysis of elementary line segments and fusion of the polylines in the previous steps. We consider either explicit or implicit polygon fusion. Explicit fusion provides the results in the form of a complete map of all free and occupied areas. Implicit fusion does not provide full information about all areas in a format that can be directly interpreted in a visual way. Its objective is to construct the additional database structures that speed up computation of the spatial queries related to free and occupied areas.

For example, the database structures that improve performance of the navigation queries are one of the goals of implicit fusion. Explicit fusion is more time consuming than implicit fusion because identification of all free and occupied areas requires multiple accesses to the database. Typically, implicit fusion requires no more than a single scan of entire database.

Our implementation of explicit fusion merges the descriptions of elementary areas into a collection of free and occupied polygons (*f-* and *o-polygons*). Informa-

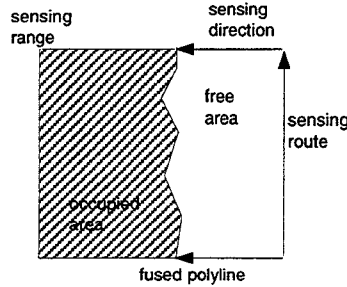


Figure 4: Elementary free and occupied areas

tion about position and shape of the elementary areas is obtained from the description of polylines, sensing routes and sensing directions. An elementary area is a pair of adjacent free and occupied areas. A free area is a polygon whose boundaries consist of a fused polyline on one side and the route of the sensing device on the other. The remaining two edges are the parallel sensing directions. An occupied area is a polygon whose boundary on one side is a fused polyline and on the other side, the maximum range of the sensing device. The parallel sensing directions form the left and right hand sides of the elementary occupied area. A sample elementary area is given in Figure 4.

The explicit fusion of *f*- and *o*-polygons starts with the merging of any two overlapping elementary areas. The result consists of one *o*-polygon and two *f*-polygons. Figure 5 presents a sample merge of two elementary areas into (i) *o*-polygon, (ii), (iii) *f*-polygons and (iv), (v) unknown areas.

The fused polygons consist of either solid or dotted edges. A solid edge represents a border between free and occupied areas, while a dotted edge represents a border between either a free or an occupied area and unknown area. Fusion of the remaining *f*- and *o*-polygons is performed by successive merges of the elementary areas with the already obtained polygons. In order to speed up identification of overlapping elementary areas and *f*- and *o*-polygons the entire space is indexed by an R-tree.

When a full map of free and occupied areas is not needed implicit fusion is more effective. In our case implicit fusion targets a class of navigational queries where an objective is to find whether a path from one location to another is entirely made up of *f*-polygons. Computation of such queries requires a number of iterations over the database contents. A single iteration starts with the identification of all *f*-polygons that contain a given point. Then, for an arbitrarily selected

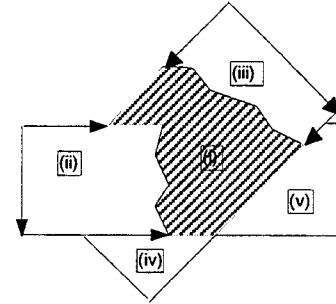


Figure 5: Merge of two elementary areas

*f*-polygon the intersection of its border and the current segment of the route is found. The intersection point is moved outside of the *f*-polygon by a small distance in the direction of the current route segment. In the next iteration a new set of *f*-polygons is found. The iterations are repeated until the destination of a route is reached or no *f*-polygon can be found on the current path.

Efficiency of such an algorithm strongly depends on how fast it is possible to find all *f*-polygons that contain a given point in a single iteration. One of the solutions would be to fuse *f*-polygons only in a given direction. In our case it is possible to fuse elementary free areas and later on use them as *f*-polygons. Correctness of such an approach is justified by the fact that the merging of free elementary areas with any other area always provides a free area of the same shape. The additional advantage of this approach is that, spatial index structures constructed for fusion of the polylines can be re-used for more effective computation of a containment test.

A procedure to find a sequence of *f*-polygons in order to detect a possible path for the robot is as follows. A grid plane whose direction vector is perpendicular to the direction of a current route segment is selected. The coordinates of the destination point are used to find the column number on a selected grid plane.

Next, the procedure finds all elementary polylines that cross the given column. An index that maps column numbers into descriptions of all elementary polylines that cross the columns allows for fast selection of relevant elementary polylines. The descriptions of elementary polylines leads to the descriptions of full polylines, sensing routes and sensing direction. From such information it is possible to fuse a description of the respective free elementary area.

A standard test that checks the number of intersection points of a current route segment with the borders of free elementary area is done to determine whether a

point is included within the free area. If only one intersection point is found we consider it as the next place on the route to which it is possible to move the robot. Otherwise, a movement is impossible and another free elementary area should be fused. If on a given plane there is no free elementary areas that contain a point then the same procedure is repeated for the next grid plane.

An index of elementary lines that cross the columns on each grid plane is used to minimize the number of containment tests.

## 5. SUMMARY

The main objective of this work was to apply the selected database mining techniques to fuse two dimensional objects from the range sensor data. We assumed that line segments are submitted in a random order by a number of robots with ultrasonic sensors. Moreover, the total number of line segments is so large that information about all of them must be stored in the spatial database. We argue that it is possible to adopt the techniques normally used for the discovery of association rules to fuse polylines and polygons from sensor data. The paper proposes organisation of the spatial database that integrates information about the sensed objects. The main contributions of the paper are the algorithms for fusion of the polylines and polygons from grid elements.

There is still a number of directions for future research. In our case organisation of the spatial database is more based on spatial clustering than on the traditional methods of spatial indexing. Efficiency of this approach should be confirmed by the experimental results. When the large number of elementary free and occupied areas are fused into *f*- and *o*-polygons the efficient merge of polygons and elementary areas is still an open problem.

## 6. REFERENCES

- [1] Agrawal R., Imielinski T., Swami A., Mining Association Rules between Sets of Items in Large Databases. Proc. of the ACM SIGMOD Intl. Conf. on Management of Data, pp. 207-216, May 1993.
- [2] Agrawal R., Imielinski T., Swami A., Database Mining: A Performance Perspective. IEEE Trans. on Knowledge and Data Engineering, 5(6), pp. 914-925, 1993
- [3] Agrawal R., Srikant R., Fast Algorithms for Mining Association Rules. Proc of the 20th Intl. Conf. on Very Large Databases (VLDB), September, 1994
- [4] Fayyad U.M., Piatetsky-Shapiro G., Smyth P.J., Uthurusamy R., Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1995
- [5] Henderson T., Weitz E., Hansen C., Multisensor Knowledge Systems: Interpreting 3D Structure. Int. Journ. of Robot. Research, 7(6), pp. 114-133, 1988
- [6] Lou R., Kay M., Multisensor Integration and Fusion in Intelligent Systems. IEEE Trans. on Sys. Man and Cyber., 19(5), pp. 901-931, 1989
- [7] Matsuyama T., Knowledge-based Aerial Image Understanding Systems and Expert Systems for Image Processing. IEEE Transactions, GE-25(3), pp. 305-316, 1987
- [8] McKerrow P.J., Echolocation - From range to outline segments. Robotics and Autonomous Systems, 11, pp. 205-211, 1993
- [9] McKerrow P.J., Robot Perception with Ultrasonic Sensors Using Data Fusion. IEEE Systems Man and Cybernetics Conf. Vancouver, pp. 1380-1385, October 1995
- [10] Nievergelt J., Hinterberger H., Sevcik K.C., The grid file: an adaptable, symmetric, multikey file structure, ACM Trans. on Database Systems, 9(1), 1984, pp. 38-76
- [11] Arlein R., Gava J., Gehani N., Lieuwen D., Ode 4.2 User Manual, <ftp://research.att.com/dist/ode>, 1996
- [12] Srikant R., Agrawal R., Mining Generalized Association Rules. Proc. of the 21st Intl. Conf. on Very Large Databases (VLDB), September 1995
- [13] Srikant R., Agrawal R., Mining Quantitative Association Rules in Large Relational Tables. Proc. of the ACM SIGMOD Conf. on Management of Data, June 1996
- [14] Shafer S., Stentz A., Thorpe A., An Architecture for Sensor Fusion in a Mobile Robot. Int. Conf. on Robotics and Automation, pp. 2202- 2011, 1986