World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:7, No:3, 2013

Spatial Query Localization Method in Limited Reference Point Environment

Victor Krebss

***Abstract*—**Task of object localization is one of the major challenges in creating intelligent transportation. Unfortunately, in densely built-up urban areas, localization based on GPS only produces a large error, or simply becomes impossible. New opportunities arise for the localization due to the rapidly emerging concept of a wireless ad-hoc network. Such network, allows estimating potential distance between these objects measuring received signal level and construct a graph of distances in which nodes are the localization objects, and edges - estimates of the distances between pairs of nodes. Due to the known coordinates of individual nodes (anchors), it is possible to determine the location of all (or part) of the remaining nodes of the graph. Moreover, road map, available in digital format can provide localization routines with valuable additional information to narrow node location search. However, despite abundance of well-known algorithms for solving the problem of localization and significant research efforts, there are still many issues that currently are addressed only partially. In this paper, we propose localization approach based on the graph mapped distances on the digital road map data basis. In fact, problem is reduced to distance graph embedding into the graph representing area geo location data. It makes possible to localize objects, in some cases even if only one reference point is available. We propose simple embedding algorithm and sample implementation as spatial queries over sensor network data stored in spatial database, allowing employing effectively spatial indexing, optimized spatial search routines and geometry functions.

***Keywords*—**Intelligent Transportation System, Sensor Network, Localization, Spatial Query, GIS, Graph Embedding.

I. Introduction

INTELLIGENT transportation systems (ITS) improve transportation safety and mobility and enhance productivity through the integration of advanced communications technologies into the transportation infrastructure and in vehicles. ITS encompass a broad range of wireless and wire line communications-based information and electronics technologies and aims to bring connectivity to transportation through the application of advanced wireless. It can be achieved by mutual interaction of nodes equipped by sensors forming a sensor network, to provide connectivity with and between vehicles; between vehicles and roadway infrastructure; and among vehicles, infrastructure and wireless consumer devices. The concept of transportation connectivity, once it has developed from research into deployment, will bring with it benefits that we are just beginning to understand:

1) A system in which highway crashes and their tragic consequences are rare because vehicles of all types can sense and communicate the events and hazards happening around them.
2) A fully connected, information-rich environment within which travelers, transit riders, freight managers, system operators, and other users are aware of all aspects of the system's performance.
3) Travelers who have comprehensive and accurate information on travel options-transit travel times, schedules, cost, and real-time locations; driving travel times, routes, and travel costs; parking costs, availability, and ability to reserve a space; and the environmental footprint of each trip.
4) System operators who have full knowledge of the status of every transportation asset.
5) Vehicles of all types that can communicate with traffic signals to eliminate unnecessary stops and help people drive in a more fuel efficient manner.
6) Vehicles that can communicate the status of on-board systems and provide information that can be used by travelers and system operators to mitigate the vehicle's impact on the environment or make more informed choices about travel modes.

Vehicular ad hoc networks (VANETs), a platform for vehicular communications, are a subgroup of mobile ad hoc networks (MANETs) with the distinguishing property that the nodes are vehicles like cars, trucks, buses or road infrastructure objects. This implies that node movement is restricted by factors like road geometry, course, encompassing traffic and traffic regulations. Because of the restricted node movement, it is a feasible assumption that the VANET will be supported by some fixed infrastructure that assists with some services and can provide access to various traffic assisting applications. The fixed infrastructure can be deployed at critical locations like slip roads, service stations, dangerous intersections or places well known for hazardous weather conditions.

Knowing the correct positions of VANET network nodes is essential to many envisioned application scenarios in the field of wireless sensor networks rely on positioning information [1]. Knowledge of location information can also improve the performance of routing algorithms because it allows the use of geo-routing techniques. Equipping all sensor nodes with specific hardware such as GPS receivers would be one option to gain position information at the nodes. However, since GPS

Victor Krebss is with the Transport and Telecommunication Institute, Riga, LV 1019 Latvia (phone: +37129243923; email: victor@cps.lv). The article is written with the financial assistance of European Social Fund. Project Nr.2009/0159/1DP/1.1.2.1.2/09/ IPIA/VIAA/006. The Support in Realisation of the Doctoral Programme “Telematics and Logistics”.

requires line-of-sight between the receiver and the GPS satellites, it may not work well indoors, underground, or in the presence of obstructions such as dense vegetation, buildings, or mountains blocking the direct view to these satellites. Another solution is to provide only a few nodes (so-called anchor or landmark nodes) with GPS and have the rest of the nodes compute their position by using the known coordinates of the anchor nodes [2].

One characteristic inherent to this approach is that the anchor density and their actual placement determine the solution quality. Obviously, in the absence of anchors, nodes are clueless about their real coordinates. The predominant type of approach, involves nodes measuring the distances between nodes themselves and their neighbors, with only some nodes called "beacons" having to be informed of their position through GPS or manual configuration.

Various new localization and spatial analysis techniques has been introduced by modern geographic information system (GIS) technologies, using digital information, for which various digitized data creation methods are used. For example road map, is a two-dimensional object that contains points, lines, and polygons that can represent cities, roads, and political boundaries such as states or provinces. GIS applications store, retrieve, update, or query some collection of features that have both non-spatial and spatial attributes.

Spatial querying capabilities can be essential for sensor network query systems. For many applications, the ability to query sensor networks in an ad hoc fashion will be key to their usefulness. Rather than re-engineering the network for every task, as is commonly done now, ad hoc querying allows the same net-work to process any of a broad class of queries, by expressing these queries in some query language. In essence, the network appears to the user as a single distributed agent whose job it is to observe the environment wherein it is embedded, and to interact with the user about its observations [3].

In this paper, we resent a GPS-free localization algorithm for wireless node localization. Proposed approach can effectively overcome the potential flip ambiguity problem, taking into consideration digital map road geometry and traffic regulations. The same principle can be applied in a 3D case. Sample implementation guidelines provided in section 5, as a spatial query over Oracle spatial database platform.

II. Related Work

The limitations of manual configuration and GPS have motivated the search for alternative ad-hoc methods, with a large number of localization systems having recently been proposed and evaluated. Recently, novel schemes have been proposed to determine the locations of the nodes in a network where only some special nodes (called beacons) know their locations. In these schemes, network nodes measure the distances to their neighbors and then try to determine their locations. The process of computing the locations of the nodes is called network localization.

Localization of nodes in VANET's, in general, can be split up into two parts: First, the process of distance estimation or measurement and second, the localization algorithm. There are different approaches for estimating the distance between a node and its neighbors or fixed anchors. Some techniques rely on the calculation of these distances with physical measurements like radio signal runtime, ultrasonic based-measurements or received signal strength indication (RSSI) measurements. Others try to approximate the distance with a hop-count indicator.

The approaches taken to solve this localization problem differ in the assumptions that they make about their respective network and device capabilities. These include assumptions about device hardware, signal propagation models, timing and energy requirements, network makeup (homogeneous vs. heterogeneous), the nature of the environment (indoor vs. outdoor), node or beacon density, time synchronization of devices, communication costs, error requirements, and device mobility.

Localization algorithms can be classified as range-free or range-based. Range-based algorithms use location metrics such as time of arrival, time difference of arrival, received signal strength, and angle of arrival to estimate the distance between two nodes. Proximity sensing between nodes is typically the basis for range-free algorithms.

Given imprecise ranging and inter-node communication, nodes scattered throughout a large volume can estimate their physical locations from a small set of reference nodes using only local information. RADAR [4], a radio frequency (RF) based system for locating and tracking users inside buildings, which operates by recording and processing signal strength information at multiple base stations positioned to provide overlapping coverage in the area of interest and combines empirical measurements with signal propagation modeling to determine user location and thereby enable location aware services and applications.

Solutions in range-free localization are being pursued as a cost-effective alternative to more expensive range-based approaches. As an example, APIT [5] algorithm requires a heterogeneous network of sensing devices where a small percentage of these devices (percentages vary depending on network and node density) are equipped with high-powered transmitters and location information obtained via GPS or some other mechanism.

However, taking in account a fact that VANET node, represented by a vehicle, operating in road physical infrastructure, it makes sense to improve localization techniques with respect to a digital map describing the road network. In this case, the geo coding facility of a Geographical Information Systems (GIS) becomes a very powerful tool to convert map location to a global (x,y) coordinate point. For this purpose, sensors detect landmarks that have been characterized in a previous passage. There are solutions [6] relying on use of additional sensors installed on board the vehicle enabling management of natural landmarks in enhanced local maps for precise localization by using single-frequency GPS data, dead-reckoned sensors and a road-map handled by a GIS.

Thus, GIS can also be used in node localization algorithms for VANETS by benefiting from the roads description stored in the map database. Unlike traditional database applications, where spatial considerations are often irrelevant, most applications of sensor networks will involve queries over spatial data.

III. Problem Formulation

The localization problem can be viewed as similar to the graph embedding problem [7]. It is clear that the immediate neighbors of the landmark can estimate the distance to the landmark by direct signal strength measurement. Using some propagation method, the second hop neighbors then are able to infer their distance to the landmark, and the rest of the network follows, in a controlled flood manner, initiated at the landmark. If a graph is sufficiently connected, and the lengths of its edges are all known, then its plane topology may be reconstructed.

Thus, localization problem can be considered as task to reconstruct the positions of a set of sensors given the distances between any pair of sensors [8] that are within some unit disk radius of each other. In this case, some of the sensors may be beacons, sensors with known positions, but results are not affected much by whether beacons are available. This problem essentially asks if a particular graph with given edge lengths can be physically realized in two dimensions.

Consider N nodes labeled $1 \ldots n$ at unknown distinct locations in some physical region. We assume that some mechanism exists through which each node can discover its *neighbor* nodes by establishing communication with those nodes, and can estimate the range (separation distance) to each of its neighbors. Each discovered neighbor relationship contributes one undirected edge $e = (i, j)$ in a graph G over the nodes.

So, we are given an N-vertex graph $G(V = \{1, \ldots, n\}, E)$, and for each edge $\langle i, j \rangle \in E$ – its Euclidean "length" $l_{i,j}$. Denote a 2D layout of the graph by $x, y \in \Re^n$, where the coordinates of vertex i are $p_i = (x_i, y_i)$. Denote

$$d_{ij} = \| p_j - p_i \| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad (1)$$

In the non-noisy version of the problem, we know that there exists a layout of the sensors that realizes the given edge lengths (i.e. $d_{i,j} = l_{i,j}$). Our goal is then to reproduce this layout. This layout is usually not unique.

Previous studies [10] have shown that the network localizability problem closely related to graph rigidity. A graph is called generically rigid if one cannot continuously deform any of its realizations in the plane while preserving distance constraints. A graph is generically globally rigid if it is uniquely realizable under translations, rotations, and reflections. Moreover, theorem [9] states:

Let N be a network in $\mathbb{R}^d$, d = 2 or 3, consisting of $m > 0$ beacons located at positions $p_{1,} p_{2,} \ldots p_m$ and $n - m > 0$ ordinary nodes located at positions $p_{m+1,} p_{m+2,} \ldots p_n$. Suppose that for the case $d = 2$ there are at least three beacons in general position. Similarly, for the case $d = 3$ suppose there are at least four beacons positioned at points in general position. Let $\mathbb{F}_p$ denote the point formation whose points are at $p_1, p_2, \ldots, p_n$ and whose links are those labeled by all neighbor pairs and all beacon pairs in N. Then for both $d = 2$ and d = 3 the network localization problem is solvable if and only if $\mathbb{F}_p$ is globally rigid.

So, all nodes in a globally rigid sub graph with at least three beacons are localizable.

Fortunately, there are additional information sources that we may exploit to eliminate spurious solutions and lack of beacons to the layout problem. Formally, we may pose our problem as follows [11]:

Layout problem, given a graph $G(V = \{1, \ldots, n\}, E)$, and for each edge $\langle i, j \rangle \in E$ – its length $l_{i,j}$, find an optimal layout $(p_1, \ldots, p_n)$ ($p_i \in \Re^d$ is the location of sensor i), which satisfies for all $i \neq j$:

$$\| p_i - p_j \| = l_{ij} \quad if \langle i, j \rangle \in E \qquad (2)$$

For the rest of this paper we assume that the sensors are embedded in the plane, namely $d = 2$. It seems that an optimal layout is unique (up to translation, rotation and reflection) in many practical situations.

However, since we aim at a distributed algorithm that should minimize communication between the sensors, dealing with repulsive forces or long-range target distances is not practical, as this will involve excessive inter-sensor interaction, which is very expensive in this scenario. To avoid this, we propose an algorithm, which is based only on direct information sharing between adjacent sensors, avoiding all communication between nonadjacent sensors or any centralized supervision.

In the real-life noisy version of the problem, the measured distances $l_{i,j}$ are contaminated by noise: $l_{i,j} = d_{i,j} + \varepsilon_{i,j}$. This means that there might not even exist a solution to the optimal layout problem.

IV. Algorithm Overview

Geo location information in this case, can be viewed in form a restrictions on the order of the edges around the vertices of graph. While it is a not so trivial task using raster geospatial data, vector type layers can provide valuable information for proper node graph embedding and orientation.

For example, shapefile [12] being a popular geospatial vector data format for geographic information systems software, describes such geometries as points, polylines, and polygons, and stores non topological geometry and attribute information for the spatial features in a data set (Fig. 1).

This data, viewed in road infrastructure context can help to improve localization process, eliminating locations where vehicle, cannot be physically present, and refine location coordinates matching vehicle location with existing roads coordinates. In that sense, roadmap data can be viewed as a graph which embraces node distance sub graph with sub graph vertices placed either on its edges or matching with its vertices.

There is another assumption can be freely made regarding road geometry. As it was mentioned above, in a vector map, a feature's position is normally expressed as sets of X, Y pairs or X, Y, Z triples, using the coordinate system defined for the map, presenting geo location road data in form of points, lines, arcs and polylines. Nevertheless, each particular data structure representation depends on digital map implementation provider and set of standard it supports.

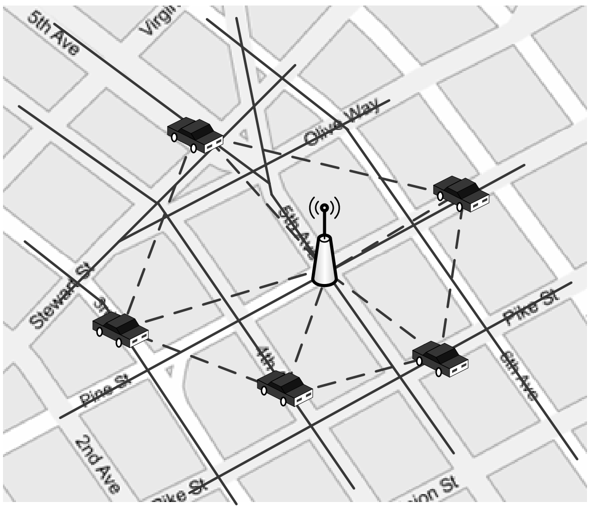

Fig. 1 Distance graph vertices embedded into road geo location road graph

Nevertheless, in our case, proposed localization algorithm needed input data is limited to road graph edges list and a binary function that can indicate if sample point belongs to particular edge or not.

Graph connectivity degree or edges shape does not interfere with algorithm logic. This fact allows significantly simplify algorithm modeling by reducing road geometry model base to grid and where each edge and sample point are arguments for intersection detecting function with certain tolerance assignable to reflect possible measurement errors, non zero road width and localized vehicle overall dimensions.

The basic idea of our iterative localization approach is to combine known neighborhood landmarks coordinates and nodes with known location (e.g., the anchor nodes) to localize others, not necessarily sufficiently connected nodes (i.e., the free nodes). Without loss of generality, we consider localization of a stationary network in a 2-D plane. We assume that the sensors are all range sensors producing distance measurements.

At least one node with known position (beacon) must be present within network to start localization algorithm. We will use a notion – neighbor, what means a network node having measured distance to the beacon node and known to algorithm executing environment. Another notion we introduce is shadow locations or shadows. It means a set of intersections of digital map representing grid and circle with center at beacon node location and radius equal to distance to neighbor. The meaning of this notion is a set of physically possible locations of neighbor node regardless of other measurements available.

Formally, the algorithm performed for each beacon node can be described as follows:

1) For each beacon node. Select all neighbor nodes (n) for this beacon and split them into node groups having measured distances to at least one other node in this group, i.e. connected sub graphs. Group number can vary from one group in case when all nodes are connected to overall number of neighbor nodes when there are no measured distances between this beacon neighbors.
2) For each group. Calculate shadow locations (m) for each node, intersecting map grid and circle with center at beacon node coordinates, radius equal to distance to from beacon to neighbor node.
3) Generate a set of all possible shadow location combinations, matching shadow nodes with any segment of map grid at given radius and preserving distances between nodes. This step has the highest computational complexity $O(n\cdot m^2)$ and determines overall algorithm complexity. However, in real life situations it should not produce large data sets.
4) If group has more than one member, test each combination comparing measured distances between nodes and distance based on grid coordinates.
5) Exclude combinations where a distances does not match, taking in account appropriate tolerance to cover measurement errors and vehicle overall dimensions. If there only one shadow location for node left, node is localized and become a beacon.

The localization problem, being similar to the graph embedding problem is strongly NP-hard [13]. As we can see algorithm at stages 4-5 in fact performs exhaustive search in order to eliminate not matching neighbor nodes combinations. Unfortunately, when the size of the instances grows the running time for exhaustive search becomes forbiddingly large, even for instances of small size.

On the other hand, with the increased speed of modern computers, large instances of NP-complete problems can be solved effectively. For example, it is nowadays routine to solve travelling salesman (TSP) instances with up to 2000 cities [14]. And if the data is structured, then instances with up to 13000 cities can be handled in practice. There is a huge gap between the empirical results from testing implementations and the known theoretical results on exact algorithms.

Moreover, described algorithm has an advantage that it is completely distributed and can be executed concurrently, with separate parts of the algorithm being run simultaneously on independent processors, and having limited or none at all information about what the other parts of the algorithm are doing. So, in a real life situation it is not likely that algorithm executing node will perform exhaustive search on such number of combination able to produce any significant delay.

V. VANET Localization as a Spatial Query

Location-based spatial queries refer to spatial queries whose answers rely on the location of the inquirer. Efficient processing of spatial queries is of critical importance with the ever-increasing deployment and use of wireless and mobile technologies. It has certain unique characteristics that traditional query processing and databases does not address.

Recently[16], there has been a growing interest in the use of location-based spatial queries, which refer to a set of spatial queries that retrieve information based on mobile users' current locations.

The wireless environment and the communication constraints play an important role in determining the strategy for processing spatial queries. This article assumes simplest approach, a user establishes a point-to-point communication with the server so that her queries can be answered on demand, and it means that operating environment contains a remote wireless information server.

As a reference implementation has been chosen Oracle Spatial, well-known integrated set of functions and procedures that enables spatial data to be stored, accessed and analyzed quickly and efficiently. Spatial data represents here the essential location characteristics of real or conceptual objects as those objects relate to the real or conceptual space in which they exist.

Oracle Spatial [15], often referred to as Spatial, provides a SQL schema and functions that facilitate the storage, retrieval, update, and query of collections of spatial features in an Oracle database. Spatial consists of the following:

1) A schema (MDSYS) that prescribes the storage, syntax, and semantics of supported geometric data types
2) A spatial indexing mechanism
3) Operators, functions, and procedures for performing area-of-interest queries, spatial join queries, and other spatial analysis operations
4) Functions and procedures for utility and tuning operations
5) Topology data model for working with data about nodes, edges, and faces in a topology (described in Oracle Spatial Topology and Network Data Models).
6) Network data model for representing capabilities or objects that are modeled as nodes and links in a network (described in Oracle Spatial Topology and Network Data Models).
7) GeoRaster, a feature that lets you store, index, query, analyze, and deliver GeoRaster data, that is, raster image and gridded data and its associated metadata (described in Oracle Spatial GeoRaster).

The spatial component of a spatial feature is the geometric representation of its shape in some coordinate space. This is referred to as its geometry.

Spatial supports the object-relational model for representing geometries. This model stores an entire geometry in the Oracle native spatial data type for vector data, SDO_GEOMETRY. An Oracle table can contain one or more SDO_GEOMETRY columns.

The object-relational model corresponds to a "SQL with Geometry Types" implementation of spatial feature tables in the Open GIS ODBC/SQL specification for geospatial features.

A common example of spatial data can be seen in a road map. A road map is a two-dimensional object that contains points, lines, and polygons that can represent cities, roads, and political boundaries such as states or provinces. A road map is a visualization of geographic information. The location of cities, roads, and political boundaries that exist on the surface of the Earth are projected onto a two-dimensional display or piece of paper, preserving the relative positions and relative distances of the rendered objects.

These applications all store, retrieve, update, or query some collection of features that have both non-spatial and spatial attributes. Examples of non-spatial attributes are name, soil type, land use classification, and part number. The spatial attribute is a coordinate geometry, or vector-based representation of the shape of the feature.

Spatial uses a two-tier query model to resolve spatial queries and spatial joins. The term is used to indicate that two distinct operations are performed to resolve queries. The output of the two combined operations yields the exact result set.

As shown in Fig. 2, the primary filter operation on a large input data set produces a smaller candidate set, which contains at least the exact result set and may contain more records. The secondary filter operation on the smaller candidate set produces the exact result set.

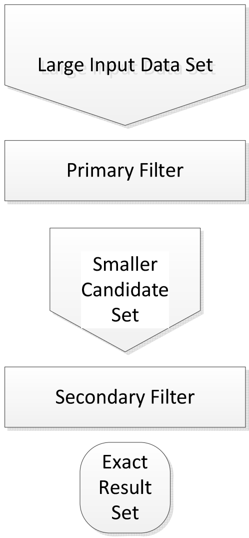

Fig. 2 Query model

The introduction of spatial indexing capabilities into the Oracle database engine is a key feature of the Spatial product. A spatial index, like any other index, provides a mechanism to limit searches, but in this case the mechanism is based on spatial criteria such as intersection and containment.

A spatial R-tree index can index spatial data of up to four dimensions. An R-tree index approximates each geometry by a single rectangle that minimally encloses the geometry (called the minimum bounding rectangle, or MBR), as shown in Fig. 3.

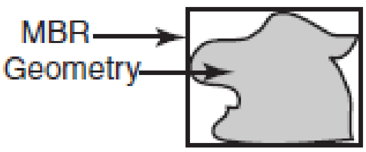

Fig. 3 MBR enclosing geometry

For a layer of geometries, an R-tree index consists of a hierarchical index on the MBR's of the geometries in the layer, as shown in Fig. 4.

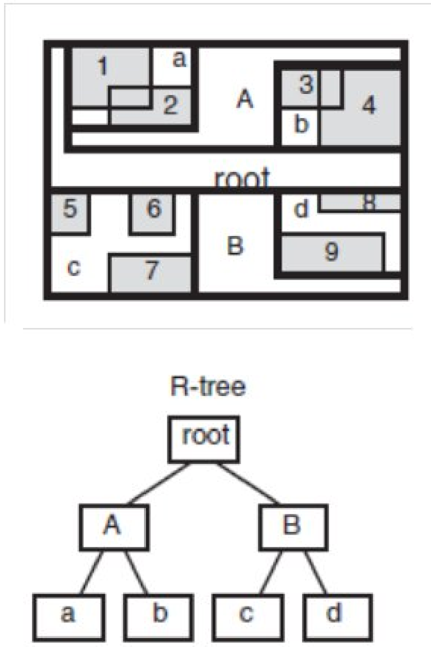

Fig. 4 R-Tree Hierarchical Index on MBR's

- *1* through *9* are geometries in a layer. *a*, *b*, *c*, and *d* are the leaf nodes of the R-tree index, and contain minimum bounding
- rectangles of geometries, along with pointers to the geometries. For example, *a* contains the MBR of geometries *1* and *2*, *b* contains the MBR of geometries *3* and *4*, and so on.
- *A* contains the MBR of *a* and *b*, and *B* contains the MBR of *c* and *d*.
- The root contains the MBR of *A* and *B* (that is, the entire area shown).

Utilizing spatial features mentioned above, let us implement algorithm described in section 4 as a spatial statements. Assume two database tables we need for this task. Each table has a column of type `SDO_GEOMETRY`. Other columns needed primarily as id numbers and needs no further explanation. We assume that first table, AREA_MAP, contain a digital map itself, beacon *b* location and known distances *D, 1..n* from beacon to nodes in form of circles *C, 1…n* with center at beacon coordinates (b_x,b_y) and radius r_i equal to distance d_i. Second table, STAGING_MAP, serves as a staging area. Spatial data indexing procedure description omitted here, since it does not interfere with algorithm logic. Nevertheless, it is worth to mention that indexing is required for optimal spatial database performance.

Whole process takes two steps, two spatial statements. First, Statement 1, finds all intersection points *V*, of circles *C* and road geometries stored in digital map, and inserts into staging area, keeping a track of to what circle c_i each particular point belongs.

Statement 1

```
INSERT INTO STAGING_MAP (STAGING_ID,
OBJECT_GEO_LOCATION, MAP_SEGMENT_ID, NODE_ID)
SELECT
DEFAULT,
SDO_GEOM.SDO_INTERSECTION
(x.segment_geo_location,
y.segment_geo_location, 0.005) GEOM,
x.segment_id,
y.segment_id
FROM
AREA_MAP x,
AREA_MAP y
WHERE
SDO_RELATE(x.segment_geo_location,
y.segment_geo_location, 'mask=ANYINTERACT') =
'TRUE'
AND x.shape_type = 'MAP_SEGMENT'
AND y.shape_type = 'NODE_CIRCLE';
```

Statement uses primary filter ANYINTERACT to narrow query window and then, SDO_INTERSECTION function performs man job, selecting intersection points. Last function parameter, tolerance, set to 0.005.

Second step, Statement 2, selects from the staging area table distinct intersection point sets, satisfying distance matrix and node to beacon distance conditions.

Statement 2

```
SELECT
S1.staging_id AS S1_ID, S1.NODE_ID AS N1_ID,
S1.OBJECT_GEO_LOCATION.SDO_ORDINATES,
…
SN.staging_id AS SN_ID, SN.NODE_ID AS NN_ID,
SN.OBJECT_GEO_LOCATION.SDO_ORDINATES
FROM
STAGING_MAP S1,
…
STAGING_MAP SN,
AREA_MAP AM
WHERE
AM.SEGMENT_NAME = 'Anchor'
AND S1.NODE_ID = :node1
…
AND SN.NODE_ID = :nodeN
/* Known distances from nodes to anchor */
AND
SDO_GEOM.SDO_DISTANCE(S1.OBJECT_GEO_LOCATION,
AM.SEGMENT_GEO_LOCATION, :tolerance1) BETWEEN
:D1 - :tolerance1 AND :D1 + : tolerance2
…
AND
SDO_GEOM.SDO_DISTANCE(SN.OBJECT_GEO_LOCATION,
AM.SEGMENT_GEO_LOCATION,
:tolerance1) BETWEEN :DN - :tolerance1 AND :DN
+ : tolerance2
/* Known distances between nodes */
AND
SDO_GEOM.SDO_DISTANCE(S1.OBJECT_GEO_LOCATION,
S2.OBJECT_GEO_LOCATION, :tolerance1) BETWEEN
:L1 - :tolerance1 AND :L1 + : tolerance2
…
AND SDO_GEOM.SDO_DISTANCE(SN-
1.OBJECT_GEO_LOCATION, SN.OBJECT_GEO_LOCATION,
:tolerance1) BETWEEN :LN - :tolerance1 AND :LN
+ : tolerance2;
```

Here, L_n stands for known distance between two nodes and D_n – distance between node and beacon. In addition, tolerance value passed to statement as a parameter.

If graph, formed by nodes, is rigid enough, or additional map information makes it rigid enough, as a result we receive one distinct point, set corresponding to ground truth nodes locations. However, in case when information is not sufficient, it is possible to receive multiple location sets, product of graph rotation for one beacon case, or graph flip for two beacons case.

VI. Conclusion

This paper proposes the concept for localizing a network of moving, range-capable nodes by location-based spatial queries for VANET environments; a method to extend the capabilities of GPS to non-GPS enabled nodes in an ad hoc network.

Positioning is based on a hybrid method combining distance vector and digital map matching and GPS or preset beacon node coordinates to estimate location in presence of signal strength measurement errors. The model evaluation confirms the applicability of the proposed approach and shows that the computational and network overheads are small.

We believe that this work is an important step towards research area. Although spatial queries have been extensively studied, to the best of our knowledge, there exists no previous work that studies distance graph matching with map regions. We expect that research interest in such queries will grow as the number of embedded or mobile devices and related services continue to increase.

References

[1] David K. Goldenberg , Arvind Krishnamurthy , Wesley C. Maness , Yang Richard , Yang Anthony Young , A. Stephen Morse , Andreas Savvides , Brian D. O. Anderson. (2005). *Network Localization in Partially Localizable Networks. in Proceedings of IEEE INFOCOM,* pp 313 – 326.

[2] Thomas Locher, Pascal Von Rickenbach and Roger Wattenhofer. (2008). Sensor networks continue to puzzle: Selected open problems. *In*

Proc. 9th Internat. Conf. Distributed Computing and Networking (ICDCN).

[3] Spatial Queries over Sensor Networks (2005). Amir Soheili, Vana Kalogeraki, Dimitrios Gunopulos. *In Proceeding: GIS '05 Proceedings of the 13th annual ACM international workshop on Geographic information systems,* pages 61 – 70, ACM New York, NY, USA 2005

[4] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher. Range-free localization schemes in large scale sensor networks. In Proceedings of the Ninth International Conference on Mobile Computing and Networking (Mobicom), pages 81–95, San Diego, CA, Sept. 2003.

[5] Paramvir Bahl and Venkata N. Padmanabhan. (2000). *RADAR: an in-building RF-based user location and tracking system*. pp. 775 – 784.

[6] Chengdu Huang , Brain M. Blum , Advsalahuddinahmed , John A. Stankovic , Tarek Abdelzaher. (2003). *Range-Free Localization Schemes for Large Scale Sensor Networks.* Department of Computer Science, University of Virginia.

[7] James Newsome, Dawn Song. (2003). Graph Embedding for Routing and Data-Centric Storage in Sensor Networks Without Geographic Information. *In SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems (2003*), pp. 76-88.

[8] James Aspnes, David Goldenberg, and Yang Richard Yang. On the computational complexity of sensor network localization. (2004*). Proceedings. Lecture Notes in Computer Science 3121*, Springer-Verlag, pp. 32–44.

[9] TRANSACTIONS ON MOBILE COMPUTING, DECEMBER 2005 1 A Theory of Network Localization J. Aspnes, T. Eren Member, IEEE D.K. Goldenberg Student Member, IEEE, A. S. Morse Fellow, IEEE W. Whiteley, Y. R. Yang Member, IEEE, B. D. O. Anderson Fellow, IEEE, P. N. Belhumeur Fellow, IEEE p-p 4–7.

[10] Yunhao Liu, Zheng Yang (2010). Location, Localization, and Localizability Location-awareness Technology for Wireless Networks. Springer New York Dordrecht Heidelberg London, pages 112-114.

[11] Craig Gotsman, Yehuda Koren. (2005). Distributed Graph Layout for Sensor Networks. Journal of Graph Algorithms and Applications vol. 9, no. 1.

[12] ESRI Shapefile Technical Description An ESRI White Paper—July 1998.

[13] J.B. Saxe. (1979). Embeddability of weighted graphs in k-space is strongly NP-hard. In Proceedings of the 17th Allerton Conference in Communications, Control and Computing, pages 480–489.

[14] Gerhard J. Woeginger. (2003). Exact algorithms for NP-hard problems: A survey, Combinatorial Optimization - Eureka, You Shrink!, LNCS, pp. 185—207.

[15] Oracle Spatial User's Guide and Reference, 10g Release 2 (10.2) B14255-03 Copyright © 1999, 2006, Oracle. All rights reserved. Primary Author: Chuck Murray.

[16] J. Zhang, M. Zhu, D. Papadias, Y. Tao, and D. L. Lee. Location-based Spatial Queries. In SIGMOD Conference, pages 443–454, 2003.