*Proceedings of the 12th International Conference "Reliability and Statistics in Transportation and Communication" (RelStat'12), 17–20 October 2012, Riga, Latvia, p. 285-295. ISBN 978-9984-818-49-8*
*Transport and Telecommunication Institute, Lomonosova 1, LV-1019, Riga, Latvia*

# VANET NODES LOCALIZATION WITH LIMITED REFERENCE POINTS AVAILABILITY

***Victor Krebss[1], Boris Tsilker[2]***

*[1]CPS Ltd*
*Skanstes 13, Riga, LV-1013, Latvia*
*Ph.: (+371)29243923. E-mail: victor@cps.lv*

*[2]Transport and Telecommunication Institute*
*Lomonosova 1, Riga, LV-1019, Latvia*
*Ph.: (+371)-67100604. Fax: (+371)-67100560. E-mail: cilkers.b@tsi.lv*

One of the major challenges in creating intelligent transportation systems has been and remains the task of object localization (locating) in motion. To some extent, it can be solved by equipping all such objects with global position system (GPS). 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 in the last decade concept of a wireless ad-hoc network (VANET). Nodes of such network are able to localize themselves being equipped with short-wave communication devices. In addition, they can use some reference points on the ground. Besides mutual exchange of information between the moving objects, such network, allows to estimate potential distance between these objects measuring received signal level. It makes possible to 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. 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.

Unfortunately, known solutions cannot simultaneously take into account many factors affecting localization quality. For example, environment affecting radio signal, no line of sight between objects, a small number of anchors, high mobility objects, etc

In this paper, we propose localization approach based on the graph mapped distances on the road map, constructed on the GIS data basis. In fact, problem is reduced to distance graph embedding into the graph representing GIS data of the area. In this case, it is possible to localize objects, even if only one reference point is available.

**Keywords**: Intelligent Transportation System, sensor network, localization, cooperative positioning, GIS, graph embedding

## 1. Introduction

Wireless vehicular communications has been identified as a key technology for increasing road safety and transport efficiency. 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 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 (the 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 neighbours, with only some nodes called "beacons" having to be informed of their position through GPS or manual configuration.

While some of such schemes are cleverly engineered, it has remained an open challenge in the field to localize each node precisely. Furthermore, most of the localization schemes determine node positions using optimisation techniques and simply assign coordinates to non-localizable nodes corresponding to a local minimum. When there are multiple configurations satisfying a given instance of the localization problem, the returned configuration may not be the one that corresponds to reality. If an erroneous configuration is used by an application, for instance event detection, then incorrect or misleading conclusions may be drawn.

In this paper, we resent a GPS-free localization scheme for node localization. Proposed approach can effectively overcome the potential flip ambiguity problem, taking into consideration ground truth road geometry and traffic regulations. The same principle can be applied in a 3D case. Algorithm can easily be turned into a distributed and executed concurrently, with separate parts of the algorithm being run simultaneously on independent processors, and having limited information about what the other parts of the algorithm are doing. Comparing to the current state of the art, proposed approach can take advantage of network dynamics as well.

## 2. 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. Localization of nodes in VANETs, 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 neighbours 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 ToA, TDoA, RSS, and AoA to estimate the distance between two nodes. Proximity sensing between nodes is typically the basis for range-free algorithms.

Worth to mention such localization examples like Recursive Position Estimation [3] providing a framework for extending position estimation throughout a sensor network. 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 modelling 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.

## 3. Method Overview

Typically [7], localization might be realized in two ways:

• Geometric methods (Trilateration, triangulation, hyperbolic methods)

• Fingerprinting methods (Signal mapping)

Geometric methods include techniques that can locate or track devices based on signal properties that are estimated. TOA/TDOA, RSS and AOA are examples of geometric measurement techniques.

Fingerprinting methods require a two-phase approach. In the first phase (also called the off-line phase), a database is formed based on signal parameters and this database is utilized in the second phase to estimate the location. Localization system needs to obtain range estimates from fixed anchors or reference points in order to estimate the location of a node. Ranges estimates can be obtained using different metrics.

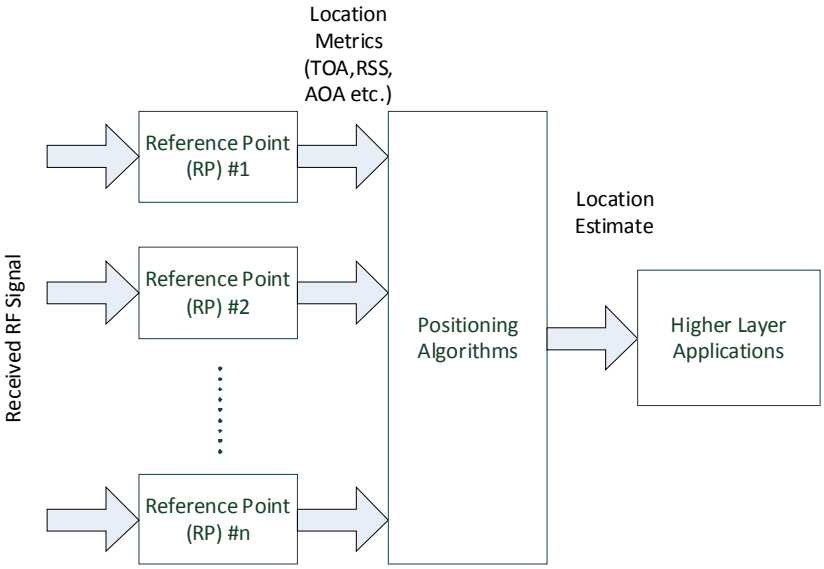


*Figure 1.* Localization system components

Figure 1 shows the components of such a localization system. The system might use different ranging metrics for obtaining the position information. RSS and TOA metrics might be considered as ranging metrics since ranging information can be obtained from these signal parameters. The nodes will need at least three ranging estimates from different anchors to be able to obtain a position fix. In the case of AOA, two different AOA estimates from two anchors will suffice to obtain a location fix.

On the other hand, the localization problem can be viewed as similar to the graph embedding problem [8]. It is clear that the immediate neighbours of the landmark can estimate the distance to the landmark by direct signal strength measurement. Using some propagation method, the second hop neighbours 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 [9] 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.

While previous approaches assumes that a non negligible fraction of nodes in the network are beacon nodes that already know their location. In contrast [10], graph embedding based approaches usually are pursuing an anchor-free localization. All nodes starts from a random initial coordinate assignment and uses only local distance estimates to converge to a coordinate assignment that is consistent with the distance estimates by exchanging only local information. The resulting coordinate assignment has translation and orientation degrees of freedom, but is correctly scaled. A post-process could incorporate absolute position information into three or four nodes (from building conventions, survey data, or GPS) to remove the translation and orientation degrees of freedom.

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 [11] 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.


*Figure 2.* Distance graph vertices embedded into road geo location road graph

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.

Figure 2 shows distance graph formed by measured distances between vehicles and between each vehicle and preinstalled road infrastructure beacon with known coordinates. This graph must be embedded into graph formed by street map such a way that all vehicles must match street graph edges. With these additional constraints vehicles can be localizes to ground truth locations when distance graph is not rigid and not sufficiently connected [12]. Moreover, non cyclic graphs (trees) still have a chance to be localized unlike in other mentioned methods. The second major advantage is that only one vehicle beacon node, or one beacon provided by road infrastructure can be sufficient for obtaining global coordinates for all localized vehicle. But of course more initial beacons, makes localization process more reliable.

While most of the existing localization algorithms for sensor networks do not consider node mobility explicitly, assuming that the network is static, VANET's show different characteristics, such as changing topology. Nevertheless, this disadvantage can be turned into a benefit when vehicles are equipped with Dead Reckoning System [13] and the distance between next and previous position of a vehicle can be computed using such movement information as direction, speed, acceleration, time, etc. Keeping in localization executing node memory previous vehicle distance to beacon node and taking in account newly measured distance as well as distance between vehicle previous and new position, distance graph can extended with more edges and vertices. This increases a chance to localize even poorly connected graph or graph with limited beacon nodes available. Extra nodes will increase graph rigidity, connectivity, add more constraints for node graph embedding within a geo location digital map and reduce possible alternative graph embedding layouts.

Using memorized vehicle locations data as additional constraints does not require proposed algorithm to distinguish between actual node location and location in past if available. A beacon node or any neighbour node directly connected to beacon can compute each known (connected) node location in iterative manner including new nodes added to graph whether it is new vehicles in line of sight or memorized existing vehicle historical location data.

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. But each particular data structure representation depends on digital map implementation provider and set of standard it supports.

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 modelling 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.

## 4. Problem Formalization

Consider $N$ nodes labelled $1,\ldots,N$ at unknown distinct locations in some physical region. We assume that some mechanism exists through which each node can discover its *neighbour* nodes by establishing communication with those nodes, and can estimate the range (separation distance) to each of its neighbours. Each discovered neighbour 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 $d_{ij}=\|p_j-p_i\|=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}$ 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. This position assignment can be unique only up to an arbitrary rotation, translation, and possible reflection, but the measured ranges determine its scale. However, for some graphs, the position assignment is not unique even up to rotation, translation, and reflection. If we treat the graph as a bar-and-joint framework, the graph should be rigid in the sense that it cannot be flexed while preserving the distances (as in a rectangle, for example). Even if the graph is rigid, it may be subject to "local flips". For example, if there are just two triangles sharing an edge, one triangle can be reflected through that edge without any distances changing. We call such a graph rigid but not globally rigid. For auto localization to work given just edge lengths, we need a globally rigid graph that has exactly one embedding.

Fortunately, there is additional information that we may exploit to eliminate spurious solutions to the layout problem.

Formally, we may pose our problem as follows [14]:

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 $i\neq j$: $\|p_i-p_j\|=l_{ij}\quad if\langle i,j\rangle\in E$.

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 a solution to the optimal layout problem might not even exist. In this case we would like to minimize the difference between the true location of the sensors and those computed by the algorithm.

## 5. Algorithm

The basic idea of our iterative localization approach is to combine known neighbourhood 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 – neighbour, 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 centre at beacon node location and radius equal to distance to neighbour. The meaning of this notion is a set of physically possible locations of neighbour node regardless of other measurements available.
The algorithm performed for each beacon node is as follows:

1. For each beacon node. Select all neighbour nodes and group by having measured distances to at least one other node in this group (connected). Group number can vary from one group in case when all nodes are connected to overall number of neighbour nodes when none is connected.
2. For each group. Calculate shadow locations for each node intersecting circle with centre at beacon node coordinates, radius equal to distance to from beacon to neighbour node and grid.

These two steps are depicted in Figures 3 and 4, where b1 stands for first beacon node, v1 to v5 neighbours forming two groups: v1, v2, v4 and v3, v5, and set of shadows from v1s1 to v5s3.

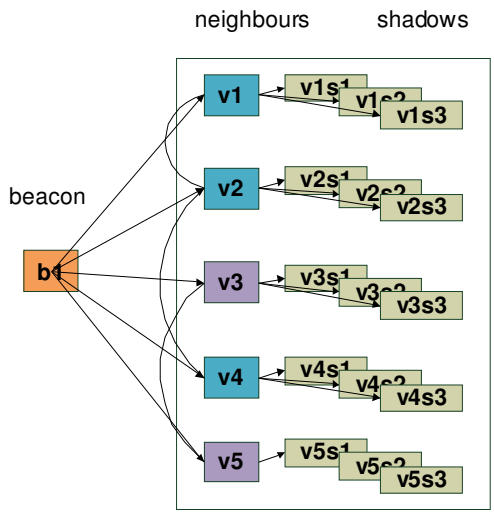


*Figure 3.* Initial algorithm data structure

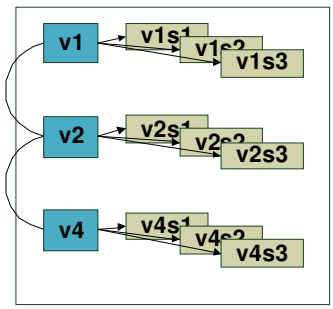


*Figure 4.* Grouped nodes

3. Generate a set of all possible shadow location combinations.

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 (Figure 5), node is localized and become a beacon (Figure 6)

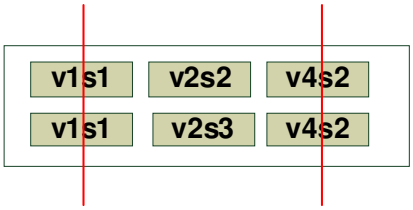


*Figure 5.* Unique location tests

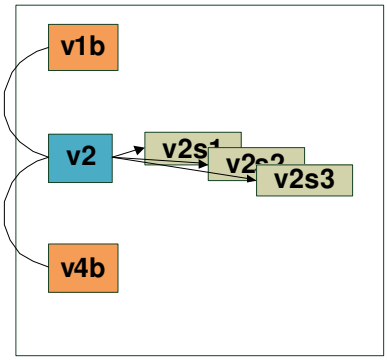


*Figure 6.* Two nodes localized

6. Algorithm repeats for each beacon node having not localized neighbours (Figure 7) until all nodes become beacons or other conditions met (maximum iterations, etc.).

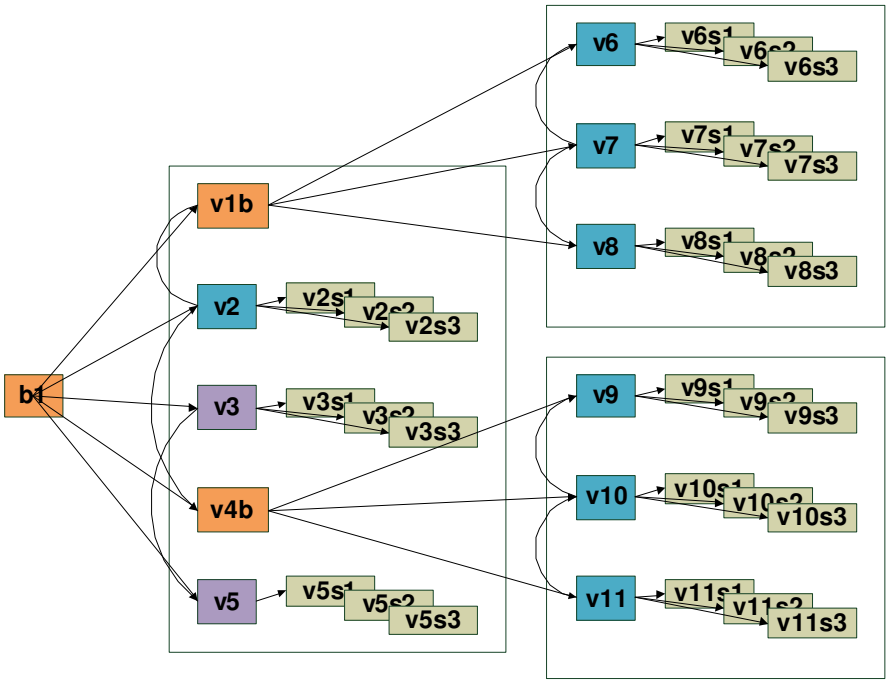


*Figure 7.* Data structure before second iteration

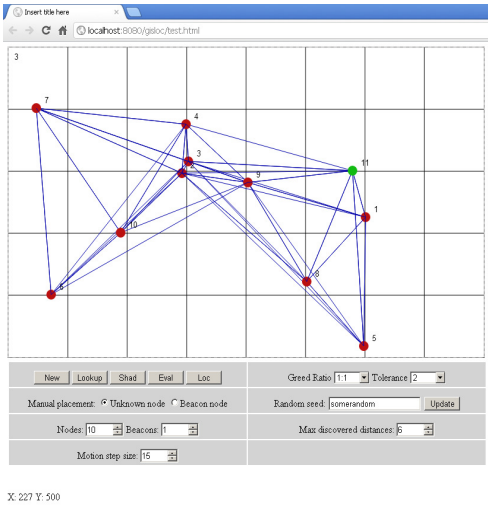


*Figure 8.* Simulation, initial state

The localization algorithm described above has been validated in simulation. Figure 8 depicts simulation environment with the following setting: ten not localized nodes each connected to six other nodes, one beacon node (11) and grid representing digital map. Figure 9 shows generated shadow locations for nodes 1, 2, 3, 5, 8, 9, after algorithm step 1 and step 2 have been executed.

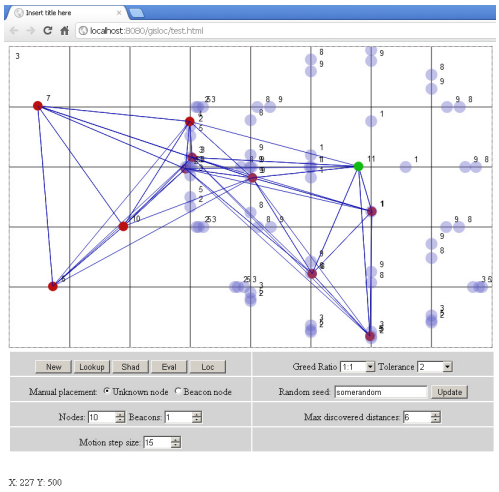


*Figure 9.* Simulation, set of nodes “shadows”

With these settings, unique possible positions were found for all six nodes within first iteration, so they were marked as localized and started to serve as beacons. At next iteration, now based on seven beacon nodes, three mode nodes (4, 7, and 10) were localized, and third, last iteration (Figure 10) gave unique position for node 6.

Considering static environment, in worst scenario, when some nodes may remain not localized, we have to stop executing algorithm when certain iteration conditions with no changes are met. However, considering dynamic environment it makes sense to continue performing localization. Nodes may change its position what gives a chance to form graph layout with no alternative embedding or establish more connections and measure distances to other nodes. If vehicles are equipped with Dead Reckoning System, it becomes possible to extend node graph with memorized previous and new, actual positions, what can significantly restrict alternative node graph embeddings. Also, other vehicles can arrive into area and significantly improve graph embedding uniqueness.

Another thing is worth to mention concerns chosen simulation model is that although digital map in form of a grid makes the model much easier to implement while remaining valid, it represents quite a

pessimistic scenario. Grid map, being completely symmetric and dense, effectively produces alternative shadow node locations and requires more beacon and higher connectivity for stable localization results. More asymmetric and less dense digital map model certainly will leave less alternative graph embedding layouts and will perform better with less beacons and lower graph connectivity.

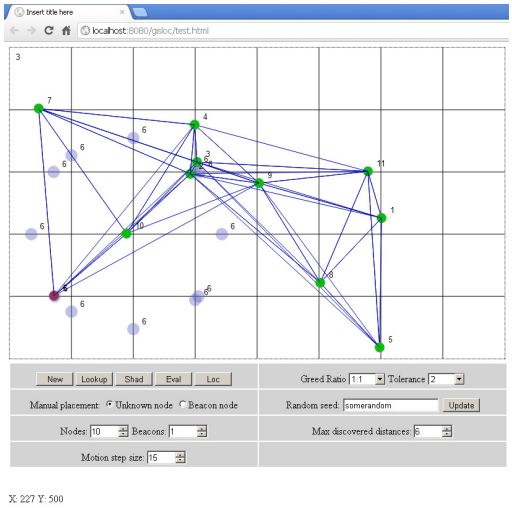


*Figure 10.* Simulation, the last node localized

Table 1 outlines the localization procedure in pseudo code. Functions and variables are named in self explanatory manner.

**Table 1**. Localization algorithm

```
while(stop conditions are not met){
Input: beacon_array[];
Input: node_array[];
for(n = 0; n < beacon_array.length; n++)
   neighbors[] = find_ neighbors(beacon_array[n], node_array[]);
   neighbor_groups[] = split_to_groups(neighbors[]);

   for(m = 0; m < neighbor_groups.length; m++){

      for(i = 0; i < neighbor_groups[m].length; i++){
         shadows[] = find_shadow_locations(neighbor_groups[m][i]);
         neighbor_groups[m][i].add(shadows[]);
      }
      shadow_combinations[] = compute_all_combinations(neighbor_groups[m]);
      possible_combinations[] = evaluate_shadow_layout();

      for(k = 0; k < possible_combinations[].length; k++){
         if(possible_combinations[k] is only one possible){
            localized_node = node_array.pop(neighbor_groups[m][node]);
            localized_node.isBeacon = true;
            beacon_array.push(localized_node);
         }
      }
   }
}

}
```

## 6. Results and discussions

The localization problem, being similar to the graph embedding problem is strongly NP-hard [15]. As we can see algorithm at stages 4-5 in fact performs exhaustive search in order to eliminate not matching neighbour nodes combinations. Unfortunately, when the size of the instances grows the running time for exhaustive search becomes forbiddingly large, even for instances of fairly 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 routine of nowadays to solve travelling salesman (TSP) instances with up to 2000 cities [16]. 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, algorithm we describe in this paper 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.

More serious challenge lies in measurement error handling and road infrastructure overall dimension tolerance. Result of matching beacon to node measured distance with grid map may depend on angle between grid segment and distance vector. When the angle is sharp, and especially tolerance or error values are significant, this may produce two or more intersections (Figure 11 a, b) mapping more possible node locations. Handling this can increase algorithm complexity and results ambiguity.

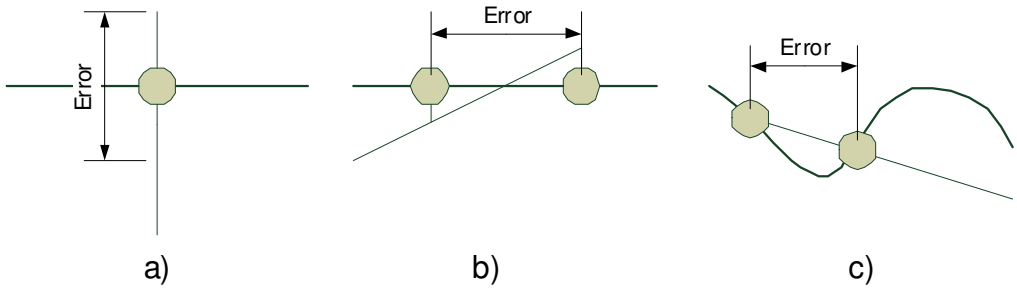


*Figure 11.*

While there are several possibilities how to modify binary function indicating if node can belong to particular map segment to take in account distance vector to path segment angle and mitigate mentioned problem, we have to keep in mind that for modelling simplicity, we have reduced geo location digital map to the grid. In a real life application, algorithm may have to deal with more complex map segments, formed by arcs, curves, polylines etc. Such type of path segments (Figure 11 c) also can produce two or even more alternative node locations fitting error vector but not realistic if we measure distance by path on a digital map.

To understand whether the above-mentioned concerns are applicable to real life applications or not, it is necessary further research and analysis of average distance errors ratio to ground truth of urban or other relevant environment on representative set of map samples.

## Acknowledgement

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" of the Transport and Telecommunication Institute).

## References

1. Goldenberg, D. K., Krishnamurthy, A., Maness, W.C., Yang, Y.R., Young, A., Morse, A.S., Savvides, A. (2005). Network Localization in Partially Localizable Networks. In Proceedings of IEEE INFOCOM, 13-17 March 2005 (pp. 313-326). New Haven, CT, USA: Dept. of Comput. Sci., Yale Univ.
2. Locher, Thomas, Von Rickenbach, Pascal and Roger Wattenhofer. (2008). Sensor networks continue to puzzle: Selected open problems. In Proc. of the 9th Internat. Conf. Distributed Computing and Networking (ICDCN), 5-8 January, 2008 (pp. 25-38). Berlin, Heidelberg: Springer-Verlag.
3. Albowicz, J., Chen, A., Zhang, L. (2001). Recursive Position Estimation in Sensor Networks. In Proceedings of Ninth International Conference on Network Protocols (ICNP'01), 11-14 November 2001 (pp 35-41). Washington, DC, USA: IEEE Computer Society.

4. Bahl, P. and V. N. Padmanabhan. (2000). RADAR: an in-building RF-based user location and tracking system. In Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00), 29 March 2000, (vol. 2, pp. 775-784), Tel Aviv, Israel.
5. He, Tian, Huang, Chengdu, Blum, Brian M., Stankovic, John A. and Tarek Abdelzaher. (2003). Range-Free Localization Schemes for Large Scale Sensor Networks. In Proceedings of the 9th ACM Annual International Conference on Mobile Computing and Networking (MobiCom 2003), 14-19 September 2003 (pp. 81–95). NY, USA: ACM New York.
6. Jabbour, Maged, Bonnifait, Philippe and Véronique Cherfaoui Heudiasyc. (2006). Enhanced Local Maps in a GIS for a Precise Localisation in Urban Areas. Intelligent Transportation Systems Conference, ITSC '06, 17-20 September 2006 (pp. 468-473). Toronto, Ont.: Univ. de Technologic de Compiegne Bonnifait, P. Cherfaoui, V.
7. Mao, Guoqiang and Baris Fidan. (2009). *Localization Algorithms and Strategies for Wireless Sensor Networks*. Information Science Reference, (pp. 57–73). Portland: Book News, Inc.
8. Newsome, James and Dawn Song. (2003). Graph Embedding for Routing and Data-Centric Storage in Sensor Networks Without Geographic Information. In Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, 05-07 November, 2003 (pp. 76-88). Los Angeles, CA, USA: ACM New York, NY, USA.
9. Aspnes, James, Goldenberg, David and Yang Richard Yang. (2004). On the computational complexity of sensor network localization. In Proceedings of First International Workshop on Algorithmic Aspects of Wireless Sensor Networks, 16 July 2004, (pp. 32-44). Turku, Finland: Springer-Verlag.
10. Priyantha, Nissanka B., Balakrishnan, Hari, Demaine, Erik and Seth Teller. (2003). Anchor Free Distributed Localization in Sensor Networks. In Proceedings of SenSys '03 of the 1st International Conference on Embedded Networked Sensor Systems, 5-7 November 2003 (pp. 340-341). NY, USA: ACM New York.
11. Environmental Systems Research Institute, Inc. (1998). *ESRI Shapefile Technical Description an ESRI White Paper*. 380 New York Street, Redlands, CA 92373-8100 USA.
12. Servatius, Brigitte, Servatius, Herman. (May, 2010). Rigidity, global rigidity, and graph decomposition. *European Journal of Combinatorics Archive,* 31(4), 1121-1135.
13. King, T., Fusler, H., Transier, M. and W. Effelsberg. (2006). Dead-reckoning for position-based forwarding on highways. In Proceedings of the 3rd International Workshop on Intelligent Transportation (WIT 2006*)* (pp. 199–204). Hamburg, Germany: Hamburg University of Technology.
14. Gotsman, Craig, Koren, Yehuda. (2005). Distributed Graph Layout for Sensor Networks. *Journal of Graph Algorithms and Applications*, 9(1), 2005.
15. Saxe, J.B. (1979). Embeddability of weighted graphs in k-space is strongly NP-hard. In Proceedings of the 17th Allerton Conference in Communications, Control and Computing, 10-12 October 1979 (pp. 480–489). Allerton House Montichello, Illinois: The Department of Electrical Engineering and the Coordinated Science Laboratory of the University of Illinois.
16. Woeginger, Gerhard J. (2003). *Exact algorithms for NP-hard problems*: *A survey. Combinatorial Optimization – Eureka, You Shrink!, LNCS,* (pp. 185-207). New York, NY, USA: Springer-Verlag, New York, Inc.


.