UDC 004.724

Nooh Taha Nasif

ALNKHBA University College, Republic of Iraq-Baghdad

ARCHITECTURE, METHODS AND ALGORITHMS OF WIRELESS NETWORK SENSORS FOR INTELLIGENT TRANSPORT SYSTEMS OPTIMIZATION

Intelligent transport systems are developing to provide effectiveness and safety of transport management with the help of information calculations and communications between transport infrastructures and vehicles. This work studies key technologies for wireless sensor networks with the aim to provide safety and connections in road networks. To solve this problem, the work uses the fact that the vehicles move along routes with a known map, which means vehicle mobility predictability and the location of the road network. With the help of statistical analysis of vehicle detection time marks, we can obtain estimates of the distances between any pairs of sensors on the roads and construct a virtual graph that is associated with the topology of the known road map on them. Wireless sensor networks are considered to be new components for intelligent transport systems and deployed on road networks to further enhance safety and protection of driving. In this article the system of wireless sensor network autonomic passive localization for intellectual transport systems, adapted and optimized for road networks is developed. The developed localization system is reliable and realistic, taking into account time synchronization error, vehicle speed deviation and various vehicular traffic intensity, as well as the lack of sensor detection and the probability of detection duplication. The APL system shows encouraging results in the localization of a road network completely covered by vehicular traffic for estimating the distance along with one vehicle speed distribution.

Keywords: *intelligent transport systems, wireless sensor networks, wireless sensors, road monitoring.*

Introduction

This work studies key technologies for wireless sensor networks with the aim to provide safety and connections in road networks [1] in such a way:

1) location of the sensor;

2) road inspection of car monitoring;

3) data transmission fir the road traffic protection;

4) reverse data transmission for information exchange about the road condition.

With this aim the road networks characteristics are researched:

1) predictable mobility of vehicles;

2) road network scheme;

3) traffic statistics;

4) vehicle trajectory.

In present times the roads are equipped with electronic sensors and variable message signs for driving, as well as GPS-based vehicle with navigation systems and systems of alarm reporting. With this tendency wireless sensor networks (WSN) can be deployed in road networks for further security enhance of traffic and mobility. US Department of Transportation and many automobile companies (for example, General Motors and Toyota) recognized promising features of vehicles (intellectual vehicles) and have recently begun to implement BSS technology for ITS infrastructures [1–3]. European of telecommunication standards institute (ETSI) applies global standards for intellectual vehicles [4], such as telematics and all kinds of connection between vehicles (for example, *«Vehicle»* → *«Vehicle»*) and between vehicles and stationary location (for example, *«Vehicle»* → *«infrastructure»* or *«infrastructure»* → *«Vehicle»*). Wireless sensor networks can be integrated in ITS for monitoring of road condition for traffic security providing (for example, highway building ground or obstructions) and announce such road condition via *«Vehicle»* → *«Vehicle»* or *«infrastructur»* → *«vehicle communications»*.

1. Formulation of the problem

For wireless sensor localization in this work autonomic passive localization (APL), intended to the road networks. It is assumed that the sensors are rarely deployed (hundreds of meters from each other) to save costs in road networks. This makes existing localization solutions based on ranking ineffective. To solve this problem, the work uses the fact that the vehicles move along routes with a known map, which means vehicle mobility predictability and the location of the road network. The binary time marks of vehicle detection and distance estimates are obtained for any pair of sensors on the roads, which allows constructing a virtual graph consisting of sensor identifiers (i.e. vertices) and distance estimates (for example, edges). This virtual graph (i.e. sensor network topology) then juxtaposed with the topology of a road map (road network), which deter-

mines places of sensors in the road networks. Many solutions were proposed for localization, using:

- accurate range measurements (for example, TOA [5], TDOA [6] and AOA [7]);

- sensor localization between sensors, Centroid [8], APIT [9], SeRLoc [10] and Robust Quads [11].

Our decision is to use vehicle on the roads as natural events for localization. The decision could be trivial, if all nodes have been equipped with vehicle complex identification sensors for localization, because distance measuring between two sensors by multiplying the average vehicle speed by the time difference of detection (TDOD) between two sensors, corresponding to the same vehicle reliably easy. Obviously, vehicle identification sensors would be costly in terms of equipment, energy and computing. The research question is how to get the location of the sensors using only the results of binary detection without the possibility of identifying the vehicle in the sensors.

With the help of statistical analysis of vehicle detection time marks, we can obtain estimates of the distances between any pairs of sensors on the roads and construct a virtual graph that is associated with the topology of the known road map on them. This display allows one to identify the sensor locations on the roads.

In particular, our localization scheme consists of three stages:

1) assessment of the distance between two arbitrary sensors in the same segment of the road;

2) construction of connectivity sensors on the roads;

3) Identification of sensors' locations by comparing the constructed connectivity of sensors with the graphic model for the roadmap.

The purpose of the article is to develop:

• new architecture of autonomous passive localization in sparse sensor networks;

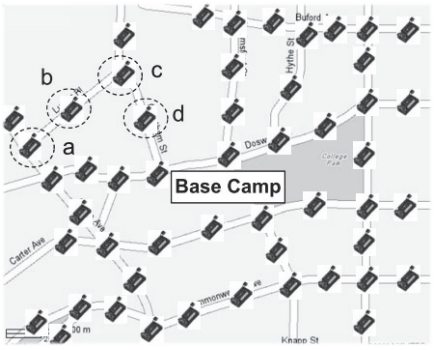
• a statistical method for estimating the lengths of road segments between two arbitrary sensors based on the time difference of detection (TDOD) concept. The TDOD operation uses a correlation between the time stamp of sensors, geographically close to each other;

• the pre-filtering algorithm to selection of only reliable estimates of the distance between two arbitrary sensors in the same segment of the road;

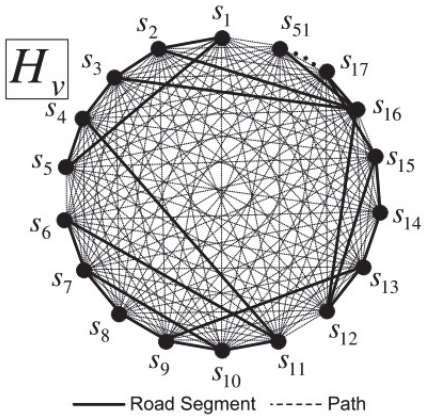
• algorithm for graph matching to coordinate the identification of the sensor with the position on the road map of the target area.

2. Basic Definitions

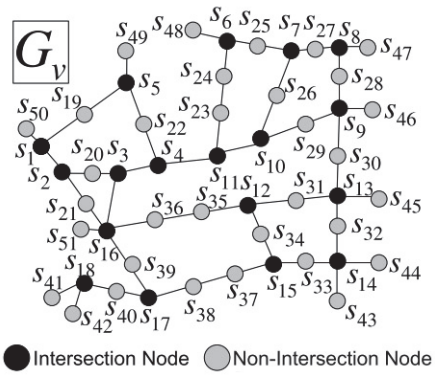
We consider a network model in which the sensors are placed both at intersection points and at points without road networks intersection. The goal is to locate the wireless sensors deployed in the road networks, only with the roadmap and timestamps for detecting binary TS, taken by the sensors, as shown in fig. 1.



a – Road network with wireless sensors



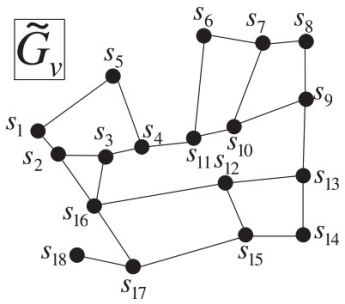
b – Virtual topology of wireless sensors: $H_v = (V_v, M_v)$



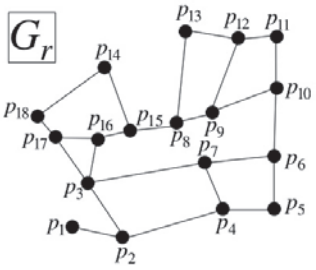
c – Virtual graph, presentive network of sensors: $G_v = (V_v, E_v)$

d – Road network only with nodes intersecting the virtual graphics

Fig. 1. Wireless sensor network deployed in the road network (beginning)



e – Reduced virtual subgraph consisting of the virtual intersection graph nodes: $\tilde{G}_v = (\tilde{V}_v, \tilde{E}_v)$



f – Real graph, corresponding to the roadmap: $G_r = (V_r, E_r)$

Fig. 1. Wireless sensor network deployed in the road network (ending)

2.1. Definitions

Define the following eight terms:

Definition 1 (intersection nodes). Sensors located at the intersection and having more than two neighbor sensors (i.e. degree ≥ 3). On the fig. 1, a sensors a and c are intersection nodes.

Definition 2 (nodes without intersection). Sensors on the road without intersection and having one or two neighbor sensors. On the fig. 1, a sensors b and d are nodes without intersection.

Definition 3 (Virtual topology). Let virtual topology $H_v = (V_v, M_v)$, where $V_v = \{s_1, s_2, ..., s_n\}$ – a set of sensors in the road network, and $M_v = [v_{ij}]$ - path length matrix v_{ij} for sensors s_i and s_j. Virtual sensors topology in the road network is shown on the fig. 1. b, fig. 1, a. M_v – full simple graph, because the edge between two arbitrary sensors exists. We define the edge of virtual topology as virtual edge. On the fig. 1, b a solid thick line represents a segment of the road between two sensors, which means that they are adjacent to the road network. A dashed thin line is an estimate of the path between two sensors, which means that they are not adjacent to the road network.

Definition 4 (Virtual graph). Let $G_v = (V_v, E_v)$ - virtual graph, where $V_v = \{s_1, s_2, ..., s_n\}$ – a set of sensors in the road network and $E_v = [v_{ij}]$ represents a matrix of the road segment length v_{ij} between sensors s_i and s_j. Fig. 1, c shows virtual graph of the net, deployed in the road network, fig. 1, a, where black node represents intersection nod, and grey nod represents a not without intersection.

Definition 5 (Abbreviated virtual subgraph). $\tilde{G}_v = (\tilde{V}_v, \tilde{E}_v)$ - abbreviated virtual subgraph, where $V_v = \{s_1, s_2, ..., s_n\}$ – A set of sensors at intersections in the road network, and $E_v = [v_{ij}]$ - road segment length matrix v_{ij} between intersection nodes s_i and s_j. Abbreviated virtual subgraph $\tilde{G}_v$ is obtained by removing nodes without intersection and their edges from the virtual graph G_v through information on the degree of G_v. For example, on the fig. 1e given virtual subgraph, consisting only of intersection nodes on the fig. 1, c is shown.

Definition 6 (Real graph). $G_r = (V_r, E_r)$ real graph, where $V_v = \{s_1, s_2, ..., s_n\}$ – set of intersections in the road network around the target area, and $E_r = [r_{ij}]$ – road segment length matrix r_{ij} for intersections p_i and p_j. The real graph can be obtained with the help of cartographic services, such as Google Earth and Yahoo Maps. A real graph, corresponding to the road network, intersection points of which have nodes of the intersection sensors are shown on the fig. 1, d. It is isomorphic to the reduced virtual graph of the graph $\tilde{G}_v$, shown on the fig. 1, e [12].

Definition 7 (The shortest path matrix). Way M – the shortest path matrix for $G = (V, E)$, such that $M_r = [m_{ij}]$ - shortest-length matrix between two arbitrary nodes i and j in G. M is calculated of E algorithm All-Pairs Shortest Paths, such as the Floyd-Warsell algorithm [13]. We define M_r as a shortest path matrix for a real graph $G_r = (V_r, E_r)$ and define M_v as a shortest path matrix for a virtual graph $G_v = (V_v, E_v)$.

Definition 8 (APL Server). Computer, which performs the localization algorithm with binary vehicle detection timestamps collected from the sensor network.

2.2. Assumptions

APL localization construction is based on the following assumptions:

• The sensors have simple sensitive devices for detecting binary vehicles without any expensive devices or GPS devices [14]. Each detection represents a set (s_i, t_j), consisting of the sensor identifier s_i and timestamps t_j.

• Ad-hocor network with tolerant delay for wireless sensors for delivery of timestamps vehicle detection to the APL server exists.

• The sensors are synchronized in time in ms. This can be easily achieved [15–16].

• APL server has information about the road map for the target zone under surveillance and can build a real graph consisting of intersections in the road network.

• Vehicle passes through all segments of the road in target road networks. Vehicle average speed is close (not identical) to the speed limit, assigned to the roads. It is assumed that the standard deviation of the vehicle speed is a reasonable value based on real traffic statistics [17].

3. APL System projecting

3.1. System architecture

We use an asymmetric architecture for localization, fig. 2 to simplify the functionality of the sensors for localization. As simple devices, the sensors only monitor the traffic and record the time of detection of the vehicle in their local stores. The APL server processes complex calculations for localization. The localization procedure consists of the following steps, fig. 2:

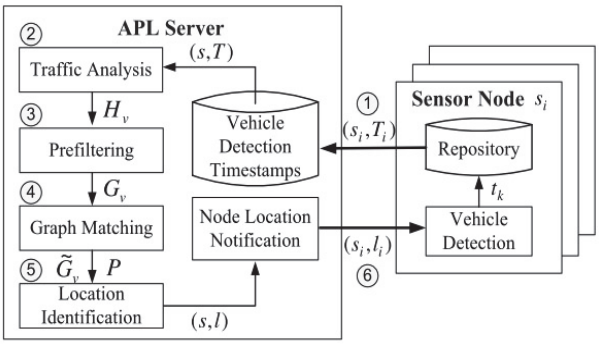


Fig. 2. APL system architecture

Step 1: After measuring the traffic, the sensor s_i sends to the server APL. Its timestamps of detection of the vehicle together with its identifier of the sensor, i.e. (s_i, T_i), where s_i – sensor identifier, and T_i – timestamps.

Step 2: The traffic analysis module estimates the length of the road segment between two arbitrary sensors with time stamp information and builds a virtual topology $H_v = (V_v, M_v)$, where V_v – set of vertex sensor identifiers, and M_v – matrix, containing an estimate of the distance of each sensors pair.

Step 3: The pre-filtering module converts the virtual topology H_v into a virtual graph $G_v = (V_v, E_v)$, where V_v – set of vertices of sensor identifiers, and E_v – adjacency matrix of the expected length of the road segment.

Step 4: The graph comparison module is constructed on the given virtual subgraph $\tilde{G}_v = (\tilde{V}_v, \tilde{E}_v)$ from the virtual graph G_r, where $\tilde{V}_v$ – a set of only nodes of intersection among V_v, a $\tilde{E}_v$ – a set of edges whose endpoints belong to $\tilde{V}_v$. $\tilde{G}_v$ is isomorphic to the real graph $G_r = (V_r, E_r)$. Then the graph mapping module calculates the permutation matrix P, making the reduced virtual subgraph $\tilde{G}_v = (\tilde{V}_v, \tilde{E}_v)$ isomorphic to the real graph $G_r = (V_r, E_r)$.

Step 5: The location identification module determines the location of each sensor on the roadmap using the permutation matrix P as to the given virtual subgraph $\tilde{G}_v$, and to the real graph G_r. Through this display, the nodes location information (s, l) is constructed in such a way, that s is the vector of the sensor identifie, and l - corresponding to the plocation vector, i.e. $l_i = (x_i, y_i)$, where s_i – sensor identifier x_i – coordinate x, y_i – coordinate y in the roadmap.

Step 6: With (s, l) APL server sends each sensor s_i to its location with a message (s_i, l_i).

Let's start from the step 2, i.e. step 1 is clear/

3.2. Step 2: Traffic analysis for estimating the length of the road segment

The closer the two sensors are located, the more the vehicle detection time marks are correlated. Therefore, we can estimate the length of the road segment by the time of travel between two neighboring sensors using the correlation of the time stamp sets of these two sensors, as well as the average speed of the vehicle on the road segment.

Time difference on detecting (TDOD) for timestamps T_i and T_j from two sensors s_i and s_j is defined as follows:

$$d_{hk}^{ij} = | t_{ih} - t_{jk} |, \quad (1)$$

where $t_{ih} \in T_i$ for $h = 1,,.,.,|T_i|$ - h - sensor timestamp s_i and t_{jk} for $k = 1,,.,.,|T_j|$ - k - sensor timestamp s_j. Let's determine TDOD:

$$\hat{d}_{hk}^{ij} = g(d_{hk}^{ij}), \quad (2)$$

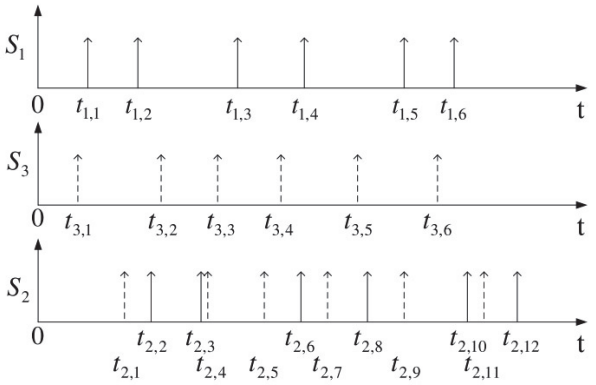
where g - the quantization function for displaying the actual value d_{hk}^{ij} to a discrete value. Number m quantization levels (i.e. q_k for $k = 1,...,m$) is determined taking into account the expected time of the vehicle movement on the longest segment of the road of the corresponding road network (1 s, 0,1 s or 1 ms).

After TDOD operation for two timestamps with the highest frequency (i.e. $\hat{d}^{ij}$) it is considered as the driving time of the vehicle for the carriageway between these two sensors s_i and s_j in the following way:

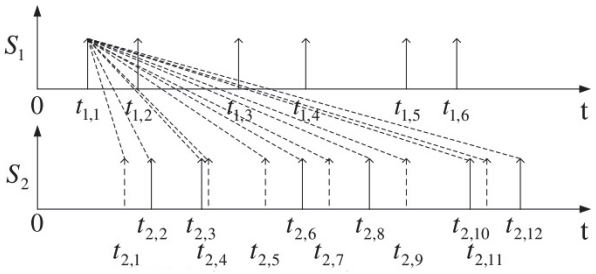
$$\hat{d}^{ij} \leftarrow \arg\max_{q_k} f(q_k), \quad (3)$$

where f – quantization level frequency q_k at $k = 1,...,m$, i.e. in (3) meaning $\hat{d}^{ij}$ is set for the quantization level q_k with maximal frequency. The driving time on the road section can be converted to the segment length of the road using formula $l = vt$, where l – segment length of the road, v – average vehicle speed, t – average time of the vehicle on the road segment.

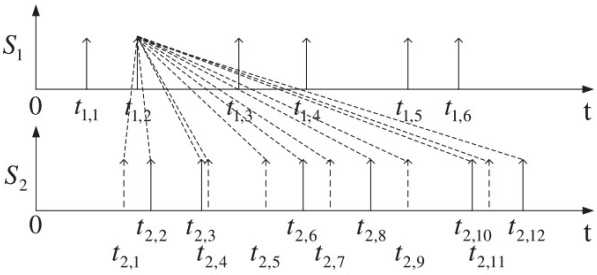
For example, on the fig. 3, a the sequence of vehicle detection at the intersection nodes is shown s_1, s_2 and s_3 on the fig. 1, f, where s_2 - a common neighbor s_1 and s_3. Fig. 3, b and 3, c show TDOD operation for nods s_1 and s_2, which is a Cartesian product for time stamps of two sets.



a – Detection of the sequence for the vehicle by sensors s_1, s_3 and s_2



b – TDOD между метками времени $t_{1,1}$ и $t_{2,i}$



c – TDOD between timestamps $t_{1,2}$ and $t_{2,i}$

Fig. 3. Time difference at (TDOD) detection

On the fig. 4 histogram [18], obtained by the operation TDOD for two timestamps is shown. Time difference value (7,3 s) with the highest frequency indicates the estimated time of travel between the two nodes.

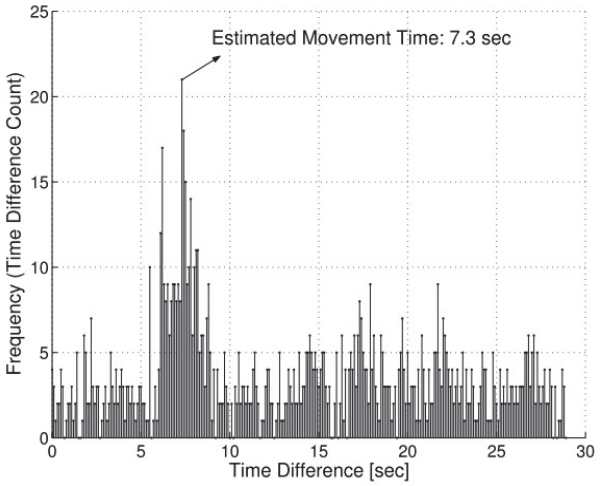


Fig. 4. Estimated travel time through the operation TDOD

We conducted an external test to check whether our TDOD operation can give good estimates for the length of the road segments in terms of vehicle travel time. The results of external testing show that our TDOD can give reasonable indicators of the length of the road segment. On the fig. 5 the test roads consist of four intersections A, B, C and D. The speed limit on these sections of the road is 64 km/h (or 40 miles/h).

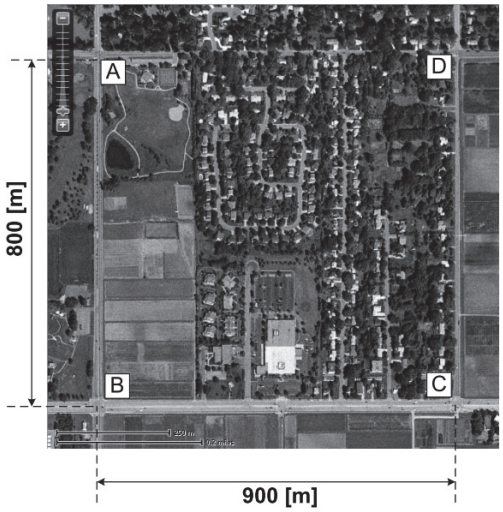


Fig. 5. Road network for outdoor testing

We conducted a manual detection of the vehicle for more accurate observation. Due to the possibility of the sensor and the physical size of the vehicle, it is difficult to obtain an accurate detection of the vehicle at intersections, so the development of an algorithm for detecting a vehicle based on its movements remains the prospect of work.

Table 1

Results of external testing

Segment of the road	Distance	Expected travel time	Measured travel time
A and B	800 m	45 s	43 s
C and D	800 m	45 s	43 s
B and C	900 m	51 s	54 s
D and A	900 m	51 s	56 s

Tabl. 1 shows the expected and measured travel time for these four road segments through the TDOD. It can be seen that the estimated travel time is close to the expected travel time. Although manual measurement can lead to some human errors, this experimental result shows substantial evidence that TDOD can provide us with estimates that are enough accurate to perform localization and obtaining edges in the virtual topology, fig. 1, b.

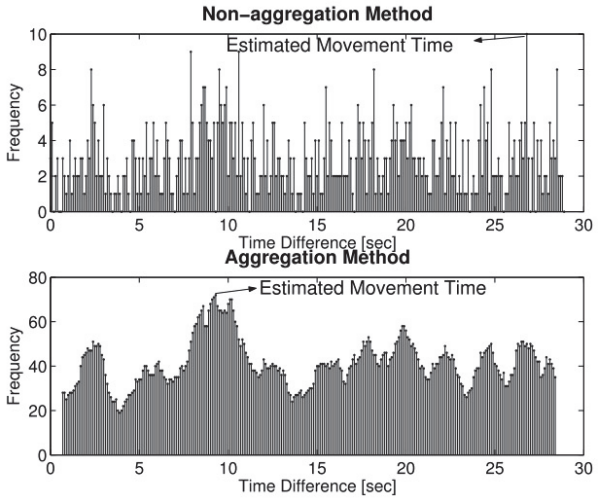


Fig. 6. Comparison between the method without aggregation and the aggregation method

Strengthening the evaluation of the road segment length

We found that an estimate close to the length of the road segment cannot always be obtained by the maximum frequency previously described by the TDOD operation. The reason is in noise estimates with higher frequencies. To solve this problem, we introduce the aggregation method when the average of several adjacent time differences becomes a new value of TDOD, and the sum of frequencies from them corresponds to the corresponding frequency. This is based on the observation that temporal differences close to the real time difference (i.e., the driving time required by the vehicle with the average vehicle speed on the road section) have relatively high frequencies by the TDOD operation for two series of time series, as shown on fig. 3. On the other hand, we observe that the noise estimate with the highest frequency occurs randomly and its neighboring estimates have relatively low frequencies. This method, based on the TDOD aggregation is named the method of aggregating and the previous simple TDOD is the method without aggregation. We determine the size of the aggregation window in proportion to the standard deviations σ_v of the vehicle speed, such as $c \cdot \sigma_v$ at $c > 0$.

On the fig. 6 comparison of the method without aggregation and aggregation method with the help of modeling is shown. The size of the aggregation window is equal to 10, So that vehicle deviation speed vehicle σ_v is 10km/h, and window size coefficient is 1. Starting from the time difference meaning, equal to 0 in the histogram for the method without aggregation we choose a representative of adjacent values of the time difference in the size of the aggregation window as the average meaning and then summarize their frequencies with the representative frequency. Then we move the window to the right by the unit of time difference and repeat the calculation of the representative and frequency. Thus, the histogram for the aggregation method is obtained by this moving window.

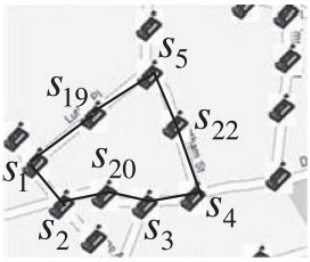
We found that for the segment of the road between the sensors s_2 and s_3 on the fig. 1, f, whose real-time difference is 9,36 s at speed of vehicle μ_v =50 km/h, method without aggregation makes an incorrect estimate (i.e. 26,8 s), but the aggregation method gives the correct estimate (i.e. 9,3 s). Thus, the aggregation method can be used to obtain good estimates of the length of the road segments in the virtual topology.

3.3. Step 3: Pre-filtering algorithm for virtual graphics

Pre-filtering algorithm is conducted for creation of the virtual graph, which has bounds estimates between virtual edges (for example, distance estimates) in the virtual topology derived from operations TDOD, Fig. 1b. Our pre-filtering algorithm consists of two pre-filters:

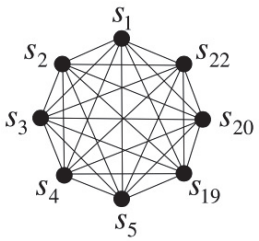
- relative deviation error;
- minimal spanning tree.

As shown in fig. 7, a, there is a partial network of roads throughout the scheme, shown in fig. 1, a, containing sensors $\{s_1, s_2, s_3, s_4, s_5, s_{19}, s_{20}, s_{22}\}$. In the virtual topology $H_v = (V_v, M_v)$. Two arbitrary sensors among them have a distance estimate, as shown in fig. 7, b. Using pre-filtering based on the relative deviation error, we remove the edges of the virtual topology that correspond to the inaccurate estimates of the trajectory, and construct a virtual graph $G_v = (V_v, E_v)$, fig. 7, c. Then we apply the preliminary filtering based on the minimal spanning tree to the virtual graph, so a virtual graph containing only edge estimates is constructed by removing the exact estimates of the path, fig. 7, d.

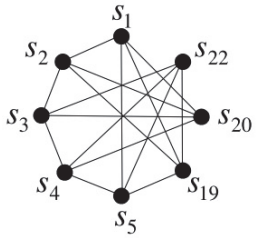


a – network with the following sensors: $\{s_1, s_2, s_3, s_4, s_5, s_{19}, s_{20}, s_{22}\}$

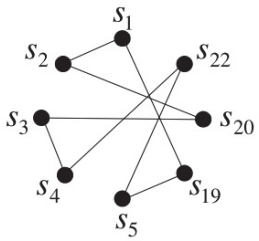
Fig. 7. The pre-filtering procedure for obtaining a virtual graph (beginning)



b – The virtual topology for the following sensors: $\{s_1, s_2, s_3, s_4, s_5, s_{19}, s_{20}, s_{22}\}$



c – The virtual graph after the preliminary adjustment based on the error of the relative deviation



d – The virtual graph after the preliminary configuration based on the minimal spanning tree

Fig. 7. The pre-filtering procedure for obtaining a virtual graph (ending)

Preliminary filtering based on the error of the relative deviation

Large errors in the path estimates will significantly affect our next steps. For example, the smallest entry in M_v must be an edge when there is no big error, because the path length is always equal to the sum of more than one edge length. However, if M_v contains huge errors, they can have any meaning in M_v. Therefore, it is very important to firstly to filter all records with large errors, treating them as estimates of the path.

We determine the relative deviation (ϕ) as a ratio of standard deviation (σ) to the average (μ), i.e. $\phi = \sigma / \mu$; Note that this is known as the coefficient of sample variation in statistics and is used to compare the variance between populations by different means. In order to calculate the average meaning μ, as well as a standard deviation σ of each record as M_v, we use several evaluation matrices M_v For the measurement time with the same duration. First, emissions from multiple estimation matrices are eliminated M_v, in order to let μ and σ to be less affected by these emissions and increase their reliability. Estimates are considered to be emissions when they are less than or greater than δ percent (for example, 20%).

To calculate the relative deviations of the estimates, we divide the vehicle detection time stamps into time windows (for example, every half an hour) and perform the TDOD operation for the time marks of two arbitrary sensors in the same time window. Then we calculate the relative deviations of the estimates of the virtual nodes for each pair of sensors. If the relative deviation is greater than a certain threshold value ε (for example, 10%), the corresponding entry is considered as an estimate of the path, and it is replaced by ∞, which indicated on the fact, that this record is a road evaluation. In the pre-filtering threshold ε is set in relation to the known deviation of the vehicle speed to the vehicle speed limit in the road network. This threshold allows preliminary filtration taking into account the actual deviation of the vehicle speed.

Prefiltration based on the minimum spanning tree

Let's suppose, that there is a virtual topology n of the sensors $H_v = (V_v, M_v)$, where V_v – set of vertices, M_v – adjacency matrix $n \times n$ of the virtual topology. Prefiltration based on the minimum spanning tree consists of three steps:

- The first step identifies the first $n-1$ edges of a virtual graph
- the second step identifies the remaining candidates to the edge of the virtual graph.
- the third step filters the path estimates between the candidate edges of the virtual graph.

Step 1: Choose $n-1$ edges from M_v, which create minimal spanning tree for a virtual topology using the minimal pairing algorithm, such as Prima algorithm [14]. It is possible to prove, that $n-1$ edges, which form MST, are boundary estimates as follows: let $M_v(i, j)$ – to be a matrix record M_v, where i – line index, and j – column index.

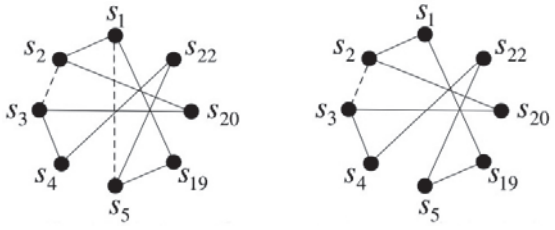
• Case 1: the smallest record must be an edge, because the path length is the sum of more than one edge length.

• Case 2: let's suppose, that we have found m edges, where $1 \le m < n-1$. Let N – be a set of corresponding nods m of edges. Then we select the smallest record $M_v(i, j)$, which satisfies $i \notin N$ and $j \in N$. $M_v(i, j)$ should be an edge for the following reason: $M_v(i, j)$ is not a rib, than another nod should exist k, for which $M_v(i, j) = M_v(i, k) + M_v(k, j)$.

1) if $k \in N$, then $M_v(i, k) < M_v(i, j)$, which contradicts our assumption that $M_v(i, j)$ – is the smallest record.

2) if $k \notin N$, then $M_v(k, j) < M_v(i, j)$, which also contradicts our assumption that $M_v(i, j)$ is the smallest.

Step 2. We find all other boundary candidates of the virtual graph $G_v = (V_v, E_v)$ from the virtual topology $H_v = (V_v, M_v)$, as shown in Fig. 1c. First, the set of edges E_v initialized to have $n-1$ ribs, obtained in the previous step. Then, with E_v, shortest path matrix M'_v is computed for the shortest path between an arbitrary pair of nodes. We use the fact that $M'_v(i,j) \geq M_v(i,j)$. For an arbitrary pair of nodes i and j $M'_v(i,j)$ – is the shortest way, created only $n-1$ edges, and $M_v(i,j)$ – those, which is created of more edges quantity; i.e. $M_v(i,j)$ may be shorter $M'_v(i,j)$. We update the set of edges E_v for M'_v, adding this new rib to E_v, and then recalculate the matrix M'_v, using E_v. We continue the process before M'_v and M_v will not be exactly the same. Thusм, we can find edges-candidates for other edges E_v from M_v.



a – Virtual graph after the step 2

b – Virtual graph after node filtration $e_{1,5}$

Fig. 8. Route filtering procedure

Step 3. We filter the path estimates between the candidate edges of the virtual graph that are not filtered out from the pre-filtering based on the relative deviation error and the pre-filtering in step 1 and step 2.

Fig. 8 shows the procedure of filtering the estimates of the path from the virtual graph after step 2. On this fig. edges $e_{1,5}$ and $e_{2,3}$ are estimates of the path in the sensor network, as shown in fig. 7, a. The idea of filtering these path estimates is to check whether there exist a path consisting of shorter edges than the path estimate or not. If there is such a path, the path estimate can be removed from the virtual graph. Otherwise, it remains in the virtual graph. For example, for the edge $e_{1,5}$ on the fig. 8, a the route, consisting of two edges $e_{1,19}$ and $e_{19,5}$, which length is shorter than $e_{1,5}$ exists. In such a way, the edge $e_{1,5}$ can be removed from the virtual graph. This check can be performed using the shortest path algorithm (for example, the Floyd-Varshall algorithm) after removing the edge $e_{1,5}$ from the virtual graph.

For the rib $e_{2,3}$ on the fig. 8b it is possible to conduct the same procedure.

Note that the candidate edge can be a real edge in the sensor network. In this case, there is ambiguity, whether this edge is a real edge or an estimate of the path. In such a way, in order accurately to conduct graph mapping, it is necessary to delete these ambiguous edges, which have alternative paths consisting of other, shorter edges of the real graph, the corresponding road network (for example, in fig. 7, a).

3.4. Step 4: Match schedules

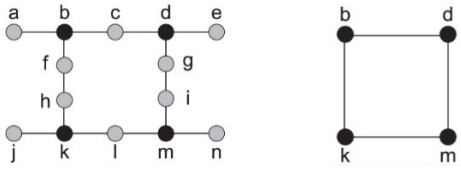
Construction of reduced virtual subgraph

Let $G_v = (V_v, E_v)$ – a virtual graph. For the isomorphic matching of graphs, two graphs must be isomorphic. Since the virtual graph G_v, returned from the pre-filtering module has more vertices and edges than the real graph G_r, we cannot directly perform an isomorphic comparison of graphs. From the observation that each crossing node has at least three adjacent sensors, the reduced virtual subgraph $\tilde{G}_v = (\tilde{V}_v, \tilde{E}_v)$ is made from a virtual graph G_v:

Let N – set of disjoint nodes G_v. Let $d_{G_v}(u)$ – level u in the graph G_v. Let e_{uv} – edge, whose ends are u and v for $u, v \in E_v$. Let $l(e)$ – be an edge length $e \in E_v$. For all $u \in N$ the following is conducted:

• If $d_{G_v}(u) = 1$, then we delete u from G_v and delete a rib, one of end of which u of G_v.

• If $d_{G_v}(u) = 1$, than we delete u of G_v, merge two edges e_{ux} and e_{uy}, one end of which u, in one edge e_{xy}. Edge length e_{xy} is set equal to $l(e_{ux}) + l(e_{uy})$.



a – Virtual graph with four intersection nodes $\{b, d, k, m\}$

b – Reduced virtual subgraph after removing nodes without intersection

Fig. 9. Construction of a reduced virtual subgraph

For example, a construction of the given virtual subgraph from a virtual graph, where a set of intersection nods is $\{b, d, k, m\}$ and a set of disjoint nodes: $\{a, c, e, f, g, h, i, j, l, n\}$ are shown on the fig. 9. After removing nodes without crossing and handling the corresponding edges, the final reduced virtual subgraph consists of four intersection nodes b, d, k, m, as shown on the fig. 9, b.

Finally, we can conduct graph mapping of the sensors net with road network, since the reduced virtual graph is isomorphic to the real graph, i.e. Two graphs have the same structure for a one-to-one mapping.

Harmonization of weighted graphs

Since the reduced virtual subgraph E_v and true graph E_r are isomorphic, our comparison of graphs can

be defined as a matrix search $n \times n$ permutations P, in order to satisfy the following: P – matrix of string permutations, and P^T – matrix of column permutations:

$$\Phi(P) = \| E_r - P\tilde{E}_v P^T \|_2^2; \quad (4)$$

$$P \leftarrow \arg\max_{\hat{P}} \Phi(\hat{P}); \quad (5)$$

$$\hat{E}_v \leftarrow P\tilde{E}_v P^T. \quad (6)$$

Let P be optimal matrix of permutations $n \times n$ (5) in terms of the minimum estimate error. Result E_v (6) is a matrix, isomorphic E_r, where the indices in both matrices indicate the node identifiers, i.e. identifier of the sensor in E_v corresponds to the intersection identifier in E_r for $i = 1,...,n$. To get an exact solution P, allowing a global minimum $\Phi(P)$, it is necessary to check all possible cases. Since this is a purely combinatorial problem, the combination-based algorithm has a time complexity $O(n!)$ for n nodes. Therefore, in reality this is an impracticable approach. We need to use approximate approaches to obtain a matrix of permutations with variable accuracy. For our goal of comparing graphs we apply an approach with its own decomposition, which has a polynomial time complexity.

We investigated the effect of the real average vehicle speed, different from the speed limit on the roads. The conclusion is that as long as all segments of the road have the same constant scaling factor for their average speeds, our localization algorithm works well, regardless of the distribution of the average vehicle velocity during the traffic measurement. In the other words, our algorithm works, although actual speeds are unknown. In case, when each segment of the road has a different scaling factor in accordance with the unbalanced congestion conditions, our algorithm does not work very well. In order to solve this problem, we propose to carry out measurements under conditions of easy traffic, for example, at night. Without congestion, we expect that all road segments have the same constant scale factor for their average speeds.

3.5. Step 5: Node location identification

Localization of intersection nodes

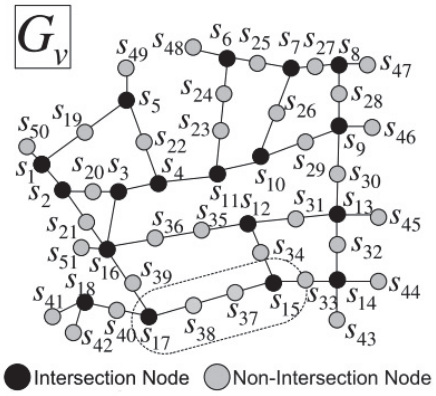
We perform the identification of the location of each intersection node with the permutation matrix P, returned from the graph matching module. Let $\sigma(s)$ – be permutation function, corresponding to the permutation matrix P, such, that $p = \sigma(s)$, where s – sensor identifier, and p – intersection identifier, corresponding to s.

$$\sigma : s \in \{1,...,n\} \rightarrow p \in \{1,...,n\}, \quad (7)$$

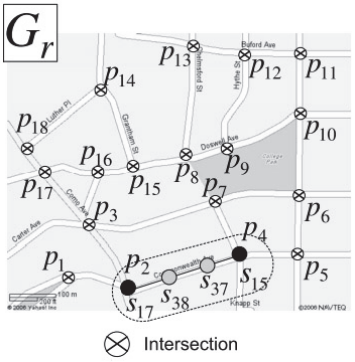
With the help of permutation function (7) we can identify the intersection identifier (p) on the roadmap for each intersection node (s).

Localization of disjoint nodes

We found the positions of the intersection nodes. Now we localize the positions of nodes without intersection of degree 2 or 1. For nodes without intersection of degree 2, using E_v of virtual graph G_v, starting from the intersection node u, and then create a route u to another intersection node v, i.e. $u \rightarrow a_1 \rightarrow a_2 \rightarrow \ldots \rightarrow a_m \rightarrow v$. All a_i for $i = 1,...,m$ are nodes without intersection, whose degrees are equal to 2. Since the nodes u and v are already localized, and all these a_i should be placed on the edge of the u to v on the given virtual subgraph $\tilde{G}_v$, as shown on the fig. 2.1e, it is possible to find statements a_i according to the length information in E_v of virtual graph G_v, as shown on the fig. 1c. We repeat this procedure, until we locate all the nodes without intersection in the virtual graph.



a – Virtual node localization graph



b – Localizing nodes without intersection in a real graph

Fig. 10. Localization of nodes without intersection

For example, on the fig. 10 localization of nodes without intersection of degree is shown 2. Two nodes without intersection s_{37} and s_{38} are located between intersection nodes s_{15} and s_{17}, fig. 10, a. We can identify the locations of the intersection nodes s_{15} and s_{17} through the localization of intersection nodes. Nodes s_{15} and s_{17} are correspondently located on the intersections p_4 and p_2. In such a way, this localization of intersection nodes allows us to know that two nodes without intersection s_{37} and s_{38} consistently placed between intersections p_4 and p_2, fig. 10, b.

For intersection nodes let w – node of degree 1. I.e. w is adjacent to the intersection node $u, v \in E_v$, then it is known which intersection w is adjacent to the

road network. For example, since nodes without an intersection s_{41} and s_{42} are adjacent to the intersection node s_{18}, then they are adjacent to the intersection p_1.

Conclusions

The developed localization system is reliable and realistic, taking into account time synchronization error, vehicle speed deviation and various vehicular traffic intensity, as well as the lack of sensor detection and the probability of detection duplication. The APL system shows encouraging results in the localization of a road network completely covered by vehicular traffic for estimating the distance along with one vehicle speed distribution.

References

1. Richard, J. (2000), Weiland and Lara Baughman Purser. Intelligent Transportation Systems. Transportation in the New Millennium, (030001).

2. General Motors (GM). Vehicle-to-Vehicle (V2V) Communications (2017), [Електронний ресурс] / Режим доступу: http://www.gm.com/experience/technology/research/overview/isl/vcim.jsp.

3. Toyota Motor Corporation (TMC). TMC Develops Onboard DSRC Unit to Improve Traffic Safety (2017), [Електронний ресурс] / Режим доступу: http://www2.toyota.co.jp/en/news/09/09/0903.html.

4. European Telecommunications Standards Institute (ETSI). Intelligent Transportation Systems (2017), [Електронний ресурс] / Режим доступу: http://www.etsi.org/WebSite/technologies/IntelligentTransportSystems.aspx.

5. Wellenhoff, B.H., Lichtenegger, H. and Collins, J. (1997), Global Positions System: Theory and Practice (4th Edition). Springer Verlag.

6. Priyantha, N.B., Chakraborty, A. and Balakrishnan, H. (2000), The Cricket Location-Support System. In MOBICOM. ACM, August 2000.

7. Niculescu, D. and Nath, B. (2003), Ad Hoc Positioning System (APS) using AoA. In INFOCOM, San Francisco, CA, USA, March 2003. IEEE.

8. Bulusu, N., Heidemann, J. and Estrin, D. (2000), GPS-less Low Cost Outdoor Localization for Very Small Devices. IEEE Personal Communications Magazine, 7(5):28–34, October 2000.

9. He, T., Huang, C., B., Blum, M., Stankovic, J.A. and Abdelzaher, T. (2003), Range-Free Localization Schemes in Large-Scale Sensor Networks. In MOBICOM. ACM, September 2003.

10. Lazos, L. and Poovendran, R. (2004), SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks. In WiSe. ACM, October 2004.

11. Moore, D., Leonard, J., Rus, D. and Teller, S. (2004), Robust Distributed Network Localization with Noise Range Measurements. In SENSYS. ACM, November 2004.

12. Douglas, B. (2000), Introduction to Graph Theory (2nd Edition). Prentice Hall, 2000.

13. Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C. (2003), Introduction to Algorithms (2nd Edition). MIT Press, 2003.

14. Gu, L. (2005), Lightweight Detection and Classification for Wireless Sensor Networks in Realistic Environments. In SENSYS, San Diego, California, USA, November 2005. ACM.

15. Elson, J., Girod, L. and Estrin, D. (2002), Fine-Grained Network Time Synchro-nization using Reference Broadcasts. In OSDI. ACM, December 2002.

16. Maroti, M., Kusy, B., Simon, G. and Ledeczi, A. (2004), The Flooding Time Synchronization Protocol. In SENSYS, Baltimore, Maryland, USA, November 2004. ACM.

17. Muchuruza, V. and Mussa, R. (2005), Traffic Operation and Safety Analyses of Minimum Speed Limits on Florida Rural Interstate Highways. In Proceedings of the 2005 Mid-Continent Transportation Research Symposium, Ames, Iowa, USA, August 2005.

18. Moore, D.S. and McCabe, G.P. (2006), Introduction to the Practice of Statistics (5th Edition). Freeman, 2006.

Надійшла до редколегії 11.09.2017

Рецензент: д-р техн. наук проф. І.В. Шостак, Національний аерокосмічний університет "ХАІ", Харків.

АРХІТЕКТУРА, МЕТОДИ ТА АЛГОРИТМИ ОПТИМІЗАЦІЇ БЕЗДРОТОВОЇ МЕРЕЖІ ДАТЧИКІВ ДЛЯ ІНТЕЛЕКТУАЛЬНИХ ТРАНСПОРТНИХ СИСТЕМ

Нух Таха Насіф

Розробляються інтелектуальні транспортні системи, що забезпечують ефективність та безпеку управління транспортом за допомогою інформаційних розрахунків та зв'язку між транспортними інфраструктурами та транспортними засобами. Бездротові сенсорні мережі вважаються новими компонентами для інтелектуальних транспортних систем і розгортаються в дорожніх мережах для подальшого підвищення безпеки та захисту водіння. У цій статті розроблена система автономної пасивної локалізації бездротової сенсорної мережі для інтелектуальних транспортних систем, адаптована та оптимізована для дорожніх мереж.

Ключові слова: *інтелектуальні транспортні системи, бездротові сенсорні мережі, бездротові датчики, моніторинг шляху.*

АРХИТЕКТУРА, МЕТОДЫ И АЛГОРИТМЫ ОПТИМИЗАЦИИ БЕСПРОВОДНОЙ СЕТИ ДАТЧИКОВ ДЛЯ ИНТЕЛЛЕКТУАЛЬНЫХ ТРАНСПОРТНЫХ СИСТЕМ

Нух Таха Насиф

Интеллектуальные транспортные системы развиваются для обеспечения эффективности и безопасности управления транспортом с помощью информационных расчетов и коммуникаций между транспортными инфраструктурами и транспортными средствами. Беспроводные сенсорные сети считаются новыми компонентами для интеллектуальных транспортных систем и развертываются в дорожных сетях для дальнейшего повышения безопасности и защиты вождения. В этой статье разработана система автономной пассивной локализации беспроводной сенсорной сети для интеллектуальных транспортных систем, адаптированных и оптимизированных для дорожных сетей.

Ключевые слова: *интеллектуальные транспортные системы, беспроводные сенсорные сети, беспроводные датчики, мониторинг дороги.*