**TRANSPORTA UN SAKARU INSTITŪTS**

**VIKTORS KREBSS**

# TELPISKO DATU BĀZU IZMANTOŠANAS IZPĒTE TRANSPORTA TIKLU OBJEKTU LOKALIZĀCIJAS METODĒS

**PROMOCIJAS DARBS**

Izvirzīts inženierzinātņu doktora zinātnisko grāda iegūšanai

Zinātnes nozare „Transports un satiksme”

Apakšnozare „Telemātika un loģistika”

Zinātniskais vadītājs:

Dr.sc.ing., B. CIĻKERS

**RĪGA – 2015**

**UDK 656:004**

**K 82**

Transporta un sakāru institūts

Krebss V.

K 82 Telpisko datu bāzu izmantošanas izpēte transporta tiklu objektu lokalizācijas metodēs. Promocijas darbs. Rīga: Transporta un sakaru institūts, 2015. 120 lpp.

**TRANSPORT AND TELECOMMUNICATION INSTITUTE**

**VICTOR KREBS**

# RESEARCH ON LOCALIZATION METHODS OF TRANSPORTATION NETWORKS OBJECTS USING SPATIAL DATABASES

**DOCTORAL THESIS**

to obtain the scientific degree Doctor of Science in Engineering

Scientific area “Transport and Communications” Scientific subarea “Telematics and Logistics”

Scientific supervisor:
Dr.sc.ing., B. TSILKER

**RIGA – 2015**

**ИНСТИТУТ ТРАНСПОРТА И СВЯЗИ**

**ВИКТОР КРЕБС**

# ИССЛЕДОВАНИЕ МЕТОДОВ ЛОКАЛИЗАЦИИ ОБЪЕКТОВ ТРАНСПОРТНЫХ СЕТЕЙ С ИСПОЛЬЗОВАНИЕМ ПРОСТРАНСТВЕННЫХ БАЗ ДАННЫХ

**ДИССЕРТАЦИОННАЯ РАБОТА**

на соискание степени доктора инженерных наук

Научная область «Транспорт и сообщение»

Научная подобласть «Телематика и логистика»

Научный руководитель:

Dr.sc.ing., Б. ЦИЛЬКЕР

**РИГА – 2015**

# ANOTĀCIJA


Viktora Krebsa disertācijas darbs „Telpisko datu bāzu izmantošanas izpēte transporta tiklu objektu lokalizācijas metodēs". Zinātniskais vadītājs, inženierzinātņu doktors, Boriss Ciļkers.

Darbā ir doti objektu lokalizācijas pētīšanas problēmu rezultāti kooperatīvajās transporta sistēmās klasisko metožu nelietojamības apstākļos.

Tēmas aktualitāte tiek izskaidrota ar bezvadu tīklu mezglu lokalizācijas metožu precizitātes un pieejamības pieaugošo nepieciešamību transporta sistēmās, transportlīdzekļu koordinātu noteikšanai, kā arī tiek izskaidrota ar transportlīdzekļu un ceļu infrastruktūras informācijas sistēmu turpmākās integrēšanas nepieciešamību.

Detalizēti tika izpētīti pašlaik pastāvošās lokalizācijas metodes, to īpatnības, pielietojamība un ierobežojumi, ko izmanto bezvadu tīklos reālajā transporta plūsmā. Īpaša uzmanība ir pievērsta scēnu analīzes metodēm, kas balstās uz distanču izmērījumiem līdz dažādiem orientieriem un uz konteksta izmērījumu zināšanām. Tiek minēti eksperimentālie dati, kuri rāda transporta stāvokļa dažādu scēnu ietekmi uz neprecizitāšu rezultējošo sadalīšanu lokalizētā objekta koordinātēs. Ir atzīmētas situācijas, kurām esot, lokalizācija ir sevišķi apgrūtināta vai nav iespējama.

Autors formulēja lokalizācijas metožu prasības kooperatīvām transportsistēmām, un ir parādīts tās realizēšanas iespējamais ceļš, kas ir paredzēts scenārijiem, kuros klasiskās metodes kļūst nepiemērotas.

Ir piedāvāta objektu lokalizācijas jauna oriģinālā metode, kas balstās uz papildus informācijas resursiem. Šajā nolūkā tiek piedāvāta jaunā lokalizācijas metode, kas balstās uz attālumu grafa izvietojumiem starp bezvadu tīkla mezgliem plaknē ar papildus, alternatīvo informācijas resursu, kā lielu ierobežojumu daudzumu, kas izslēdz alternatīvos izvietojumus, piesaistīšanu. Parādīta gan šīs metodes „tīrās" algoritmiskās realizēšanas, gan arī kompaktās, optimizētās realizēšanas principiālā iespēja, telpisko pieprasījumu pret ģeoinformācijas sistēmām veidā.

Izstrādāts imitācijas analītiskais telpiskais modelis, kas ļauj izpētīt un validēt piedāvāto metodi. Izpētīta metodes realizēšana dažādos, reāliem scenārijiem maksimāli pietuvinātos izvietojumos, ir savākti, apstrādāti, izanalizēti un novērtēti eksperimentālie dati.

Iegūtiem rezultātiem ir universāls raksturs, un tie rāda, ka piedāvātā metode var būt pielietota kā trūkstošās informācijas avots citām lokalizācijas metodēm, vai kā laicīgais informācijas avots klasisko lokalizācijas metožu darbnederīguma periodā.

## ANNOTATION


The thesis of Viktors Krebss (Victor Krebs) "Research on localization methods of transportation networks objects using spatial databases". The scientific supervisor is Dr.sc.ing., Boris Tsilker.

The main goal of the research is to explore the possibility of using alternative information resources in localization tasks, providing decision-making support for cooperative transportation systems, in cases when classical localization methods are not applicable.

Relevance of the topic is explained by the increasing need to improve the accuracy and availability of the wireless networks nodes localization methods in transportation, as well as the need for further integration of information systems of vehicles and road infrastructure.

The research considers existing localization methods, their characteristics, applicability and limitations imposed by the use of wireless networks in real traffic. Particular attention is paid to the scene analysis methods based on measured distances to various landmarks and knowledge of the measurement. Provided experimental data showing the influence of the different transportation situation scenarios on the resulting distribution of errors in the coordinates of the localized object. Especially noted situations where localization becomes very difficult or impossible at all.

Author formulates requirements for transport systems cooperative localization methods and shows a possible way to implement them, taking in account scenarios when the classical methods are not applicable.

As a result, author proposes a new original approach to object localization, relying on additional information resources. To accomplish this, new proposed localization method based on the embedding of the graph distance between the nodes of a wireless network on the plane with additional, alternative information resources as a set of constraints that exclude alternative graph realizations.

Proposed method has been realized both as "pure" algorithmic implementation, and as a compact, optimized implementation in the form of spatial queries to the Geographic Information System. A simulation-spatial analytical model has been developed to allow investigate and validate the proposed method. Implementation of the method is studied at different, close to the actual embedding scenarios. Data, collected within experiment cycles, has been processed, analyzed and evaluated.

Main results of the thesis presented are universal in nature and shows that the proposed method can be widely used as a source of missing information for other localization methods, or as the source of information for the period of time when the classical methods of localization are unavailable.

# АННОТАЦИЯ

Диссертационная работа Виктора Кребса «Исследование методов локализации объектов транспортных сетей с использованием пространственных баз данных». Научный руководитель доктор инженерных наук, Борис Цилькер.

В работе представлены результаты исследования проблем локализации объектов в кооперативных транспортных системах в условиях неприменимости классических методов.

Актуальность темы объясняется возрастающей необходимостью повышения точности и доступности методов локализации узлов беспроводных сетей в транспортных системах для определения координат транспортных средств, а также необходимостью дальнейшей интеграции информационных систем транспортных средств и дорожной инфраструктуры.

Детально исследованы существующие на сегодняшний момент методы локализации, их особенности, применимость и ограничения, накладываемые использованием в беспроводных сетях в реальном транспортном потоке. Особое внимание уделено методам анализа сцен, основанным на измерениях дистанций до различных ориентиров и знании контекста измерений. Приводятся экспериментальные данные, показывающие влияние различных сценариев транспортной обстановки на результирующее распределение погрешности в координатах локализованного объекта. Отмечены ситуации, при которых локализация исключительно затруднена либо невозможна.

Автором сформулированы требования к методам локализации для кооперативных транспортных систем и показан возможный путь их реализации, рассчитанный на сценарии в которых классические методы становятся неприменимы.

Предложен новый оригинальный метод локализации объектов, опирающийся на дополнительные информационные ресурсы. Для этого предлагается новый метод локализации, основанный на размещения графа расстояний между узлами беспроводной сети на плоскости с привлечением дополнительных, альтернативных информационных ресурсов в качестве множества ограничений, исключающих альтернативные размещения.

Показана принципиальная возможность, как «чистой» алгоритмической реализации данного метода, так и компактной, оптимизированной реализации в виде пространственных запросов к геоинформационным системам.

Разработана имитационно-аналитическая пространственная модель, позволяющая исследовать и валидировать предложенный метод. Исследована реализация метода на различных максимально приближённых к реальным сценариям размещения, собраны, обработаны, проанализированы и оценены экспериментальные данные.

Полученные результаты носят универсальный характер и показывают, что предложенный метод может быть применен в качестве источника недостающей информации для других методов локализации, либо как в качестве временного источника информации на период неработоспособности классических методов локализации.

## ACKNOWLEDGEMENT

The thesis is written under the financial aid of European Social Fund. Project Nr. 2009/0159/1DP/1.1.2.1.2/09/IPIA/VIAA/006 (The Support in Realization of the Doctoral Program “Telematics and Logistics” of the Transport and Telecommunication Institute).

# CONTENTS

# ABBREVIATIONS

| | |
|---|---|
| ITS | Intelligent Transportation System |
| V2V | Vehicle to Vehicle |
| V2I | Vehicle to Infrastructure |
| I2I | Infrastructure to Infrastructure |
| LBS | Location Based Services |
| GNSS | Global Navigation Satellite System |
| A-GPS | Assisted Global Positioning System |
| DGPS | Differential Global Positioning System |
| RSS | Received Signal Strength |
| AoA | Angle of Arrival |
| ToA | Time of Arrival |
| TDoA | Time Difference of Arrival |
| QoT | Quality of trilateration |
| GIS | Geographic Information System |
| API | Application Program Interface |
| Ad hoc | for this |
| VANET | Vehicular ad hoc Network |
| TIS | Target In Sector |
| NLOS | Non Line of Sight |

# LIST OF ILLUSTRATIONS IN THE BODY OF THESIS

# LIST OF TABLES IN THE BODY OF THESIS

# INTRODUCTION

## 1. The relevance of the problem and motivation of the research

Ensuring the ever-increasing requirements for sustainability and competitiveness of European mobile transportation systems creates the need for certain innovations. The most promising type of such innovations are Intelligent Transport Systems (ITS) that combine the most advanced solutions in the field of information and communication technologies [1, 2, 3 and 4]. The potential of ITS capable of accomplishing a wide range of tasks related to the transport sector in all its diversity [5, 6]. Suffice it to mention the electronic payment system for transport, dynamic traffic management, including adaptive speed limits, parking assistance, support for navigation applications in real-time, operational information transmission and processing for stability control and collision avoidance systems. This goes far beyond the range of tasks assigned to ITS. In addition to the marked, ITS designed to improve the interaction between different transport modes in multimodal transport systems [7, 8].

It is obvious that to achieve these goals one of the main requirements for ITS is the high degree of communication development that allows to maintain intensive information flows specific operational decisions support.

Significant expansion of the range of applications covered by the ETS, and reducing development costs can be achieved integrating information systems of vehicles and road infrastructure. Open information systems, united in one standard platform can ensure compatibility and interoperability between the infrastructure systems and equipment. This should accelerate the development of cooperative systems, based on the exchange of information between vehicles and between vehicles and the road infrastructure [9, 10].

In this regard, it should be noted separately two interrelated tasks that require new and more effective solutions. The first of these is the development of new cooperative systems and their evaluation from the standpoint of the possibility of establishing a unified approach to the implementation strategy in ITS. And the other, related to first one - definition of specifications and standards for communication in cooperative systems such as “infrastructure to infrastructure” (I2I), “vehicle to infrastructure” (V2I), “vehicle 2 vehicle” (V2V), taking into account that the objects in both cases are regarded as the wireless network nodes [11, 12, 13, 14, 15, 16, 17, 18, 19 and 20].

Until now issues related to capacity and effectiveness of applications aimed at improving traffic safety (communication type V2V) and similar issues related to applications that focus on optimizing traffic flow (communication type V2I) remains open.

One of the central and indispensable elements in the majority of cooperative ITS applications is localization (position estimate) of vehicles [21, 22, 23 and 24]. Any measurement data lose their value without knowledge of the exact coordinates of the place where they were received. Generally, localization methods are used in wireless networks to determine the coordinates of nodes using the a priori known coordinates of the individual components and the results of some measurements, such as distances, angles, and other connectivity. It should be noted that the methods of localization in wireless networks is not a trivial GPS methods extension or methods based on radars.

Localization of objects in the ITS-based wireless networks have a number of features [25, 26, 27, 28, 29 and 30], such as a variety of types of measurements, work in difficult conditions, where it is impossible to provide line of sight between nodes, limited communication and computing capabilities of nodes and dynamically changing external environment and network configuration. However, before the actual localization must determine whether the node is localizable, i.e., whether there can be found a unique coordinates showing the position of the node in the area, on the basis of available data.

Reliable information about the coordinates of the nodes of wireless networks in the ITS opens up new possibilities in the development of a rapidly growing segment of applications based on location (Location Based Services, LBS) [31, 32, 33, 34 and 35].

Despite the undeniable progress in the localization methods that are using together the data received from the GPS and built-in sensors of different types, still there are situations when the availability and accuracy of localization are not high enough. Accumulated errors in localization system based on motion sensors and the mirroring effect of a signal in an urban environment adds an error component unacceptable for many modern applications what requires further search for opportunities to improve the accuracy and reliability of the localization methods.

Significant improvement of the results of localization could be achieved employing digital maps of the area and opportunities for visual recognition of landmarks. The problem here is that the segmentation and recognition of landmarks is still extremely difficult and inadequately costly process. In this case seems promising multilevel data summarizing schemes, which uses the data obtained from different independent sources. However, the creation of such systems highlights the new challenges associated with synchronization, resource management, and the need for high computing power [36, 37 and 38].

In practice, there may be situations where none of the wireless nodes has no information about its coordinates [39, 40]. For some applications, knowledge of absolute position is not critical, and can be limited by the relative coordinates of the nodes in the network. Moreover, improved methods of relative localization may provide a means to improve the methods of absolute localization. In this case, the assignment of all network nodes virtual coordinates in Euclidean space is a special case to solve the problem of graph realization in the plane, which is a daunting task requiring a new, more effective decisions.

In recent years, a number of studies were devoted to the localization of wireless networks in order to find new, efficient and stable algorithms that take into account the mobility of the nodes, as well as a shortage and unreliability of information resources. This indicates that the problem of localization remains an open challenge requiring continuing search for solutions. This fact was the motivation for the present work aimed at studying the possibility of using new, additional sources of information for locating objects in the cooperative transport systems, in cases where classical methods are not applicable.

## 2. The methodology and methods of investigation

In recent years, researches in the field of cooperative systems have focused on improving the accuracy of localization as a critical problem. Attention is paid to this problem in the European Action Plan to support the deployment of ITS (Roadmap to a Single European Transport Area) [41]. It is particularly important to have an accurate real-time data in collision avoidance systems, navigation systems, and control systems. It is assumed that the required improvements can be achieved by multi-modal localization and by combining different types of measurements from different data sources. Despite the fact that the development of such methods has achieved a certain progress, there are still situations where localization accuracy decreases to unacceptable limit. Moreover, modern localization systems do not provide sufficient reliability and allow a situation where localization becomes impossible.

As an example can be mentioned differential GPS (DGPS) and the assisted GPS (A-GPS) - two of the most advanced technology GPS, which allow highly accurate position and high speed determining the object. Nevertheless, the use of the GPS receiver as the only source of data for locating the vehicle may become unreliable, particularly in densely built urban areas where the satellite signal is distorted significantly. Literature [42, 43, 44, and 45] describes a number of solutions that offer information to supplement GPS measurements with dead reckoning, which allows improving the accuracy and reliability of localization.

Dead reckoning (DR) - a method of determining the next position of mobile object in a series of short time intervals, based on the known direction of travel, speed, and data on the previous position. The method is simple, but it leads to incremental accumulation of errors and requires frequent resetting at short intervals to perform correction. One method of correction is to use GPS measurements to eliminate the accumulated DR errors in the moments when measurements are available.

Another approach (scene analysis) involves the detection and recognition of targets on the ground. This allows a comparison of the DR and GPS data with data from spatial databases. In particular, [46, 47, 48 and 49] in theirs researches indicates that additional information can be obtained either by using a laser scanner for recognizing the pavement edge or a range finder which can measure the range to detected targets. Further, the data received from the DR, GPS and Geographic Information System (GIS) are joined and processed in order to provide more accurate information about the location of the object, even under conditions of significant GPS signal distortion. Given that the GIS data stores digitized roadmap, buildings plan and other landmarks, the stage of training for these systems is usually not required.

Since digital maps can provide a large amount of data about the environment, the visual data may also find use in the localization methods. However, as noticed [47], due to the fact that the image processing requires extensive computational resources, GIS databases, as a rule, stores and recognizes only certain key images. Thus, at each moment, the uncertainty of individual data sources can be analyzed and compensated using the error covariance matrix that ensures the least mean square error values.

Another promising approach is the cooperative localization, assuming the presence of several vehicles connected by wireless means of communication. As an example, one can consider positioning system indoor [50]. In this case, the data necessary for localization may be supplemented not only by various types of measurements but also by measurements taken on different network nodes. This allows keeping the availability of nodes localization, either indoors or even underground that is, not having visibility of GPS satellites. In this situation, the network nodes that have their own location data are so-called “anchors” for the other nodes in the network.

Despite some success in developing methods of cooperative localization, a number of questions remains open. For example, the same set of measurements can allow multiple network configurations, thus existing localization methods can return incorrect data about the coordinates of the network nodes. In addition, most techniques of localization based on the optimization techniques, simply assigns to not localizable nodes coordinates local minimum. All this can lead to incorrect operation of the application, based on the coordinates of the nodes in the network

Using the scene analysis algorithms along with other methods of localization allows obtaining the location and eliminating erroneously detected topological network configuration. Especially effective is an iterative implementation of both these methods. In this case, the scene analysis algorithms are close to the pattern recognition problems, in which the observed physical characteristics are compared with the data and reference images available localization system.

Usually, the localization methods based on the scene analysis come down to two basic steps: collection of the area information and the position estimate based on the measured physical parameters. The two most common approaches in the evaluation of the coordinates are not parameterized statistical method of k-nearest neighbor (kNN) and probabilistic methods.

KNN method involves the received signal strength preliminary data collection and creation of the corresponding database, and then, in the process of localization, it involves the database search for closest data corresponding to observed data by the least-squares error. A probabilistic approach implies the possibility of the existence of n locations for the observed set of measurements, calculating the probability according to Bayes formula.

There is another approach successfully working in indoors environment developed for Bluetooth networks and presented by [51]. The device sends queries to other network members with different power levels, and depending on the response of other mobile nodes, estimates a relative distance. This method requires at least one “anchor” node with known coordinates, and provides accurate localization up to 1.88 meters.

Important factors that affect the quality of the scene analysis localization methods are the density and location of the networks “anchors”. This especially concerns algorithms, which uses the measured distance between the mobile nodes. At the same time, there are additional sources of data that can significantly improve localization solutions. Since the advent of GIS, it has been presented many new methods of spatial analysis based on digital maps. GIS applications allow to store, modify and query both spatial and non-spatial attributes for a particular position.

Spatial queries to the GIS can be an important tool for managing wireless networks formed by vehicles. Such queries can be adapted for each specific task, thereby avoiding the need for changes in the network configuration. In essence, such a network should become a single agent distributed to the user, delivering data from the region of interests as needed.

Thus, localization algorithms can use these digital maps available through GIS, as a set of constraints and can be reduced to the placement of the graph given in the form of a matrix of known distances between members of the cooperative transport system, and that was the main idea of this study. It should be emphasized that the proposed localization methods in are not universal for any possible scenario, however, can successfully complement the classic methods of localization and replace them in an environment where, due to lack of information resources classical methods are not applicable.

## 3. The object and subject of research

The objects of research are methods of localization of mobile objects in transportation systems, organized based on wireless networks.

The subject of the study is to investigate the possibility of localization of mobile objects of transport systems in environments of the limited availability of spatial data.

## 4. Research goals and objectives

The goal of this work is to study the possibility of using alternative information resources for decision-making support in cases when classical localization methods of cooperative transport systems objects are not applicable.

To achieve this goal, the following objectives have been formulated:

1. Review and classification of localization methods used in modern cooperative transportation systems;
2. Identify the main problems associated with localization in cooperative transportation systems, and formulate the factors that contribute to the lack of information resources;
3. To analyze the applicability of the existing classical methods and their combinations in scenarios with limited access to information resources;
4. To propose and develop an alternative method for localization based on spatial queries to geographic information systems, used as an additional source of data;
5. Build a model based on spatial database, which provides supports digital maps fragments, placement of wireless nodes at random coordinates with the given parameters, the ability to set the conditions of the line of sight between nodes, specifying the coordinates of “anchors”, as well as support for spatial queries and indexing spatial data;
6. Carry out numerical experiments for different localization scenarios and check the validity of prepositional method. Visualize the results and analyze them;
7. Present the evidence-based recommendations for the use of the method to decision-making support in cases when classical localization methods of cooperative transportation systems are not applicable.

**5. Hypotheses of the study**

The study presents the following hypothesis:

1. Placement of the graph distance between nodes of the cooperative transportation system with additional geographic information resources can serve as a means of localization in conditions where other methods are inapplicable;
2. The results of the localization obtained by the proposed method can be used as the source of the missing information for the classical methods of localization or as a temporary source of information for the period of inoperability of the classical localization methods.

**6. The methodology and methods of investigation**

To achieve the goals set in this research, were used methods of computational geometry, probability theory and graph theory, as well as the theory of simulation systems and combined analytical and simulation. The research is based on materials of scientific thematic publications, collections of articles of international conferences, the European Commission reports, the results of scientific research papers and monographs.

To test the hypotheses and the proposed methods numerical experiments were carried out. Appropriate software was created based on spatial database Oracle Spatial 11g and software platform Java. Calculations and visualization carried out in a Microsoft Office Excel 2010. The data obtained were used in several cycles of research, in particular evaluation of relevance, context research for the development of the following requirements to the application, the definition of eligibility criteria and evaluation of the results, according to the procedure described in [52].

## 7. Scientific novelty

The scientific results obtained in the thesis are as follows:

1. Proposed a GPS independent approach to the localization of the nodes of wireless networks in transportation systems using alternative information resources;
2. Developed and implemented the algorithm for GPS independent localization task using a matrix of measured distances between objects and vector digital map of the area as an additional set of constraints to exclude alternative object placements;
3. Proposed a compact and effective implementation of the method using spatial queries to the database of indexed geolocation data. Implemented analytical and simulation models;
4. Performed validation of the model, investigated the scope of the method and possible scenarios for its use.

## 8. The practical value and implementation

The results showed that the proposed localization method could be used:

1. As an independent method, in cases when for various reasons the classical methods based on the measured angles and distances to points with known coordinates cannot be applied;
2. To compensate temporary dysfunction of other localization methods due to external interference and other obstacles to obtaining the necessary measurements;
3. For data correction in localization systems that utilizes integration of data obtained from several sources (data fusion).

Specified the opportunities of method expansion for three-dimensional localization support and compatibility with any type of digital maps, supporting vector representation of the data.

Developed models implemented as a set of programs on Oracle Spatial platform 11g, allows simulating different scenarios to investigate the localization and the effectiveness of the method in various areas of transport infrastructure configurations.

## 9. The theses submitted for defense

Within the research, following theses has been formulated:

1. Localization of objects in the wireless transportation network using GPS devices as a single data source can become not reliable enough, especially in the densely built-up city. Using geolocation context and integration with other data sources can substantially compensate this disadvantage;
2. There is a need to develop new and improve existing centralized localization algorithms that are resistant to lack of information resources. Algorithms should provide absolute and relative location estimation and remain functional with both static and mobile objects. As the basis of these algorithms is advisable to use scene analysis methods;
3. If you have a matrix of the measured distances between objects wireless vehicles network, real coordinates of at least one of the objects of the network or transportation infrastructure and digital maps, localization task can be reduced to the distances graph placement on the plane, given the constraints of the digital map. Moreover, the matrix range measurements may not be complete;
4. Placement of distance graphs in the plane can be performed algorithmically, based on the Euclidean coordinates propagation, limiting the alternative placements of non-rigid graph fragments by digital map vector data;
5. The proposed localization method can be implemented using spatial queries, allowing to make it compact and to take advantage of the out of the box advanced technology of spatial indexing, and compatibility with existing digital geolocation data formats;
6. Employing proposed algorithm and additional data sources allows compensating the possibility of temporary GPS dysfunction, as well as, if necessary, correcting the errors encountered by other localization methods.

## 10. Approbation of the research

The basic concepts and findings were reported and discussed at scientific conferences and seminars. The author has 9 publications on the subject of the thesis including indexed at peer reviewed scientific literature citation databases such as Index Copernicus, Google Scholar, EBSCO, SCOPUS, ERA, DOAJ and ProQuest Index. The research results have been used in the project COST IC0906 “Improving the accuracy of real-time positioning of moving objects in mines”.

**11. Thesis structure**

The thesis consists of an introduction, five chapters, conclusion, bibliography, and three appendixes. The paper contains 120 pages, and includes figure 52 and 15 tables. Bibliography includes 151 references. The structure of the work is as follows.

*Introduction* devoted to examine the relevance of the thesis research; there are formulated goals and objectives of the study, described scientific novelty and practical value of the results.

*First chapter* discusses three important problems in the development of Location Based Services for intelligent transport systems: localization of wireless network objects, localizability and network coverage evaluation. Conclusions are made that, despite of significant progress in the localization methods development, based on the combined use of different sensors and GPS, still there are situations in which these tools and methods are ineffective or do not apply at all.

*Second chapter* describes the classification of localization methods and the types of measurements required for the implementation of the algorithms. Closely reviewed algorithms that are using as an input conditional distance determined by the number of transit transitions between nodes. In addition, this chapter pays attention to the algorithms that are more accurate, requiring measurement of Euclidean distances, angles, time, signal propagation delay and other physical parameters of the environment based on optimization techniques, numerical methods and the methods of computational geometry.

*Third chapter* gives the characteristic of network nodes localization errors resulting from inaccuracies in the measurement of distances to reference points. The influence of the relative placement of reference points on the resulting error localization. The created model and a number of experiments carried out to investigate the properties of the probability distribution of the resulting error localization depending on the probability distributions of the error distance to reference points. Tested behavior with not only a conventional normal distribution but also with the exponential distribution, and some combinations of different types of error distributions.

*Fourth chapter* summarizes in practice the general and theoretical problems of wireless sensor networks localization task. Pointed out possible directions in solving such problems in cases where the classical methods of localization are not applicable. Proposed the conceptual idea of novel localization method involving additional information resources. Developed a model, used for a preliminary verification of the algorithm and its performance evaluation, as well as computational complexity and possible deficiencies that requires further development.

*Fifth chapter* shows a possible implementation of the localization algorithm for wireless networks with additional information resources. Closely reviewed and defined features of the implementation and customization of software solutions based on spatial queries. Chapter shows the model building principles, which enables different localization scenarios simulation. Specific scenarios and parameters have been selected for the study and analysis of the algorithm in this particular implementation. Conducted a series of experiments, collected and processed data, more accurately characterized the range of arising problems and opportunities for real life application of the proposed algorithm. Usage constrains of such application also considered.

*Conclusion* presents an overview of the most significant results of the thesis, discussion and recommendations on opportunities to further develop the proposed solutions.

*Appendix* contains a list of publications of the author, the key specification of spatial functions and data structures used in the implementation of the localization method using spatial queries and the main classes of localization applications based on spatial queries.

# 1. LOCALIZATION OF OBJECTS IN INTELLIGENT TRANSPORTATION SYSTEMS AND RELATED TASKS

This chapter discusses the general problem of telematics in the context of intelligent transportation systems (ITS), where mobile and stationary objects form Ad-hoc wireless networks. It marks a special role of such networks in the implementation of applications involving knowledge of the location of mobile objects (Location Based Services) and the consequent problem of localization. Outlines the basic concepts of the localization methods and related problems of determining network localizability. In conclusion considers the most common problems that arise in the development of methods of localization, such as the assessment of network coverage, to provide an accurate localization requirements related to the specifics of particular applications and tasks that require new or more effective solutions.

## 1.1 Intelligent Transport Systems and Telematics

Telematics, as an interdisciplinary domain of study Computer Science and Technology, includes telecommunications, transportation technology, safety issues, the provision of information services, and related engineering tools. In general, tasks of telematics can be reduced to the development of transmission technologies, receiving, processing and storage of data to monitor and manage remote objects.

Of particular note are applied problems of telematics in the transport sector, especially the problem of positioning and tracking of vehicles using global navigation systems and mobile networks, as well as wireless communication between vehicles to exchange information on the location, speed, dangerous situations on the road, etc.

Such communication involves organizing temporary Ad-hoc local wireless networks and should be based on wireless devices and sensors, installed in vehicles [53, 54]. In addition, wireless devices can be installed at fixed locations and transport infrastructure that adds new features such networks and allows them to establish connections with other similar networks or the global network. The most widely represented realizations of such networks operate at short range, in UHF & VHF bands and using protocols IEEE 802.11 [55, 55]. Applications requiring longer-range work with infrastructure networks WiMAX (IEEE 802.16) [57, 58], GSM [59, 60] or 3G [61, 62].

Concept of vehicles interaction using wireless networks, as we move from the research stage to the development stage, enables creation of sophisticated intelligent transportation systems (ITS) that can perform tasks such as:

- Notification systems, that helps to reduce the risk of accidents;

- Fully-connected road environment in which drivers, freight managers, system operators and other stakeholders are aware of all aspects of the system in a timely manner;
- Provide accurate and timely information for transport operators about transit opportunities and schedules, prices, parking, routes and positioning of vehicles in real time;
- Access for the system operators to the most complete knowledge of the state of each element of the transport system.

Possibility of communication of all vehicles within systems what allows to adjust the motion that will eliminate unnecessary stopping and optimize schedules and fuel consumption. Although, in general, the ITS are mentioned in the context of all kinds of vehicles, but the EU Directive 2010/40 / EU of 7 July 2010 clearly sets out the main principles of the implementation of intelligent transportation systems in road transport and interfaces with other modes of transport. The directive defines ITS as systems in which information and communication technologies are applied in the field of road transport, including infrastructure, vehicles and users, and formalizes the movement governance and mobility, and interfaces with other information systems [63].

Cooperative communication on the road includes the mutual communication between vehicles and between vehicles and infrastructure. Data received from vehicles by the infrastructure devices usually sent to the centralized server for joint processing. These data can be used to detect a variety of events, such as changing weather conditions or vehicles log jams. The server in turn, can send recommendations as for individual vehicles, as well as for groups, enabling road safety.

The definition of cooperative systems of the European Commission reads is as follows: “Operators of traffic, infrastructure, vehicles, drivers and other road users should be cooperated for more efficient, safe and comfortable traffic. Cooperative systems such as car-to-car and car-to-infrastructure should contribute to the solution of these problems by autonomous systems” [64].

## 1.2 Transport Ad-hoc networks, VANET, V2V and I2V

The nature of problems solved in transport systems involves the use of decentralized self-organizing wireless networks whose nodes can independently move in all directions, closing and establishing connections with neighboring nodes. Such networks are known as Mobile Ad-hoc Network (MANET - Mobile Ad hoc Network). As a basic type of such networks for ITS considered Ad-hoc network for vehicles (VANET - Vehicular Ad-hoc Network), a standard for being developed under the working group IEEE 802.11p [65, 66].

Communication type Vehicle-to-Vehicle (V2V), provided by such networks, allow dynamic and anonymous data exchange between neighboring vehicles. These communications make it possible to discover (at least) the relative positioning of vehicles, their speed, and with the appropriate applications to recognize dangerous situations, calculate risks and propose certain actions to prevent them.

In general, V2V communication are provided by a basic application capable of transmitting and receiving messages like "I'm here", containing information which allows to estimate the location of the source of the message. This message may be generated using external location-based sources, such as GPS, or based on vehicle internal sensor data, allowing determining the location from the speed data, acceleration, and taking in account relative location to other vehicles.

Since the data for the messages: "I'm here" can be obtained not only by the built in car devices, development and implementation of V2V applications could be more rapid and independent from the manufacturers of automotive equipment and embedded systems [67].

Furthermore, the possibility of V2V communication itself can be regarded as an additional sensor of a vehicle that can increase the number of data sources, and allows looking beyond both space and time, making possible the creation of an entirely new class of applications. Cars can serve as sources of information, its consumers and, if necessary as data relaying entities. The necessary information can be obtained from the preset infrastructure devices that via communication of type Vehicle-to-Infrastructure (V2I) extends the capabilities of applications far beyond conventional embedded sensors.

However, despite the considerable amount of research work in the field of VANET and ITS applications, practical implementation faces a number of challenges that have only partial solutions. Among these problems is the lack of a unified technology platform capable of supporting all the necessary applications. Regardless of the chosen technology factors, affecting its competitiveness remains unchanged: the cost (the cost of equipment, maintenance and the cost of transmitting data units), quality (bandwidth, latency, and scalability), and accessibility (coverage area and performance, including an indoor environment [68].

Solving these problems requires standardization with mandatory involvement of the automotive industry. It is necessary to adopt standards that are aligned with technical solutions that have been successfully implemented and used by manufacturers. The use of a single standard for transmitting and receiving equipment, the specification of the level of transport communication (Fig. 1) in the model of OSI (Open Systems Interconnection) [69] allow solving such problems as geographical addressing, multiple message routing, security and management of communication channels.

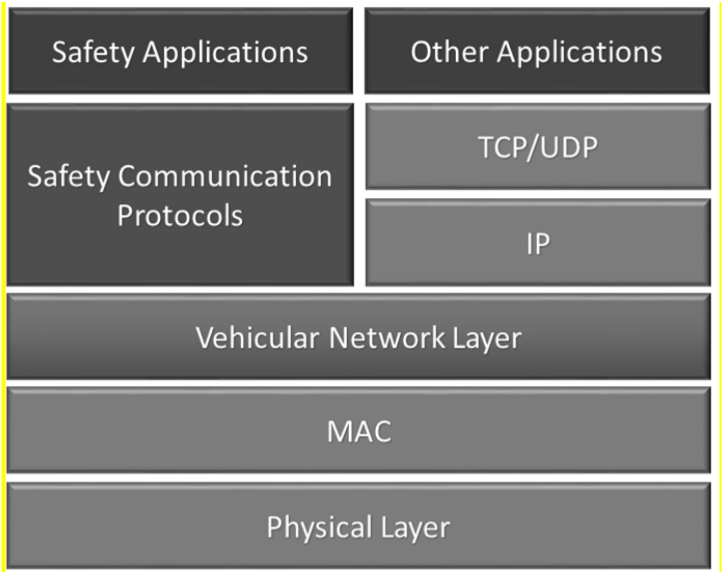



***Fig. 1.*** *Transport communication in the OSI model*

Thus, the integration of VANET solutions into a single technical platform will provide new opportunities for the transport industry, taking into account the requirements of services, mobile communication features and properties of the network members. Because of this, a special place in such a platform should be given to the spatial and temporal relationships between traffic participants (vehicles) and the context in which the application is running [70].

### 1.3 Location Based Services and localization

Key and becoming increasingly common technology platform for the implementation of applications in vehicles Ad-hoc networks are Location-based services (LBS), services based on knowledge of the location. Supporting a wide range of communication types, from short-range Bluetooth, to global satellite technology (Fig. 2), LBS are designed to provide a basic understanding of the circumstances and the spatial context of the problem being solved by application. Localization is the process of computing the location of wireless devices on the network, localizability, as an answer to the question of whether the participant network can be uniquely localized, and evaluation of network coverage are the three essential components of a technological platform [71].

In simple terms, the fundamental problem of localization can be formulated as the problem of estimating the coordinates of a point in two-dimensional or three-dimensional space coordinate system formed by some other reference geometry data. As a rule, the calculation of the new coordinates is carried out taking into account the well-known fact of the previously known coordinates of the point changes. In the case of wireless networks, the presence of a device capable of transmitting and receiving signals is assumed, which in a greater or lesser extent involved in the localization process.

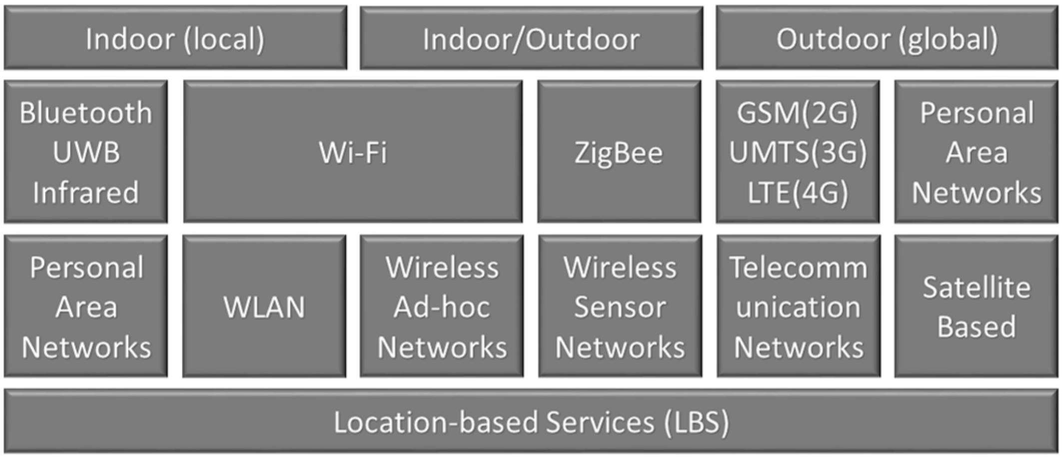



***Fig. 2.*** *LBS technologies*

Modern localization systems still relies on the basic data such as distances and angles, which may be obtained through indirect measurement by the related parameters, such as signal strength, signal arrival time, signal arrival time difference, phase difference and other data. New technologies allowed not only to obtain more accurate results, but also to create new architectural localization systems solutions, such as cooperative systems and data fusion systems.

Advantages, features and limitations of each technology-specific localization always depends on the acceptable cost of the solution, the number of localized objects, existing technologies, the level of autonomy and other properties of the application. A well-designed application must comply with an adequate balance between the requirements of the system, technological capabilities and related costs [72].

Object localization methods in wireless networks are typically used for estimating the location of the sensor with an initially unknown coordinates with respect to some of the sensors, whose coordinates are known a priori and are available for the necessary inter sensor measurements. Sensors with a priori known coordinates are called anchors. The anchors coordinates can be obtained using global positioning systems or as a result of the installation of the anchor in geodetically certain point.

In applications that require the presence of the global coordinate system, these anchors will determine the global coordinates of the wireless network. If knowledge of the physical location of the object is not critical for application and local coordinate system is sufficient, network objects are localized only in relation to each other. Then, the location data of the network typically is a product of rotation, reflection or transformation of the real global location [73].

### 1.4 Localizability

Localizability of wireless network characterizes the possibility of unique localization in the chosen system of coordinates of all nodes in the network. Network localizability analysis is means that having a set of network nodes with known coordinates (anchors), a set of nodes whose coordinates are unknown, and the measured distances between a set of nodes, to determine whether it is possible to calculate the coordinates of all non-localized nodes. This fundamental algorithmic problem in the theory of wireless networks is the subject of a large number of recent studies. For example [73, 74].

The natural follow-up question that arises solving of this problem is whether the possible network location unique. Network with a given set of known coordinates of nodes and a set of known distances between nodes is considered uniquely localizable in the case when for the initial data there is a unique set of coordinates of all nodes in the network. Unique network localizability in two-dimensional space can be described by the graphs rigidity theory. Unique localizability depends only on the combinatorial properties of the network and is completely determined by the network graph distances and anchor nodes set. Edges of the graph in this case correspond to the distances, and vertices to the network nodes. Vertices of the graph are connected by an edge if the distance between the nodes of the network is explicitly known.

Based on the technique of measuring the distance, the actual location of the Ad-hoc wireless network can be modeled as a graph of distances $G = (V, E)$. Where a set of vertices $V$ is a set of wireless communication devices in the network, and a set of edges $E$ represents the set of distances between the communications devices $(i, j)$, in the case where the distance between points $i$ and $j$ may be measured, or both have known coordinates of the vertices and are anchors. For each edge, we can use the function $d(i, j)\colon E \to R$ to get measured distance values between $i$ and $j$.

Within this model, it is necessary to answer the question of whether the resulting graph distances representing the wireless network is localizable. What means that for each vertex $v$ of the graph $G$, given the set of possible constraints $H$ (known coordinates of the anchors vertices), there exists a unique location $p(v)$ such that $d(i, j) = ||p(i) - p(j)||$ for all pairs $(i, j)$ in the set of edges $E$, and respecting the set of constraints $H$, where $||\ .\ ||$ denotes the Euclidean distance on the two-dimensional plane.

As can be seen, in contrast to the localization, which purpose is to determine the coordinates of wireless network nodes, the localizability problem focuses on determining of unique localization possibility and is a fundamental and important part of the localization process. Localization algorithms often involve the use of a large amount of computing resources and make sense only in the case where localization a priori is possible. Thus, verification of the localizability before performing localization algorithm avoids unnecessary and costly computations.

In addition, information about localizable networks can be of great importance in the operation of the network and network management, including topology control, network deployment, mobility monitoring and geographic routing. For example, planning network placement on the ground, the localizability test results may be taken into account for the verification and finding the optimum placement, number and density of anchor nodes.

Although the problem of determining network localizability became relevant mainly due to the Ad-hoc wireless sensor networks prevalence, the problem of the unique of graphs placement on the plane attracts researchers in various fields over 30 recent years.

### 1.5 Coverage estimation in VANET networks

In addition to the problems of tracking, location and location calculation, one of the fundamental problems in wireless VANET networks is the problem of network coverage estimation. Because of the large diversity as sensor types and applications, the problem of network coverage may have different interpretations and is generally defined as the wireless sensor networks measure of quality of service estimation.

Such estimation allows determining the weaknesses in the distribution of static sensors and finding optimal network configuration. It uses two approaches: the evaluation of the worst cover when the estimation is focused on identifying areas where network coverage may not be available and the best coverage estimation when the greatest importance is given to the calculation of the areas in which the coverage quality is guaranteed at the required level. From a conceptual and algorithmic point of view, the coverage estimation is defined as a method for estimating the coverage and the corresponding optimal proven estimation algorithm.

Despite the fact that the coverage estimation algorithms well studied, usually, they take into account only the uniform (isotropic) sensor model. In practice, the coverage range of the sensor in different directions often is substantially different. This means that the maximum allowable distance between the sensors is largely dependent on their orientation, and such a sensor network is called anisotropic. Such a model is more realistic, because most of the sensors, such as cameras, directional microphones, radar, etc. are anisotropic. Efficient and accurate assessment of the coverage in an anisotropic network remains an open problem in the field of wireless sensor networks [75].

**Coverage estimation methods**

### 1.5.1 Geometric method

In contrast to isotropic, directional sensor cannot cover the area over the entire circumference and has a certain sector of coverage. Therefore, coverage area can be considered as a sector on the two-dimensional plane. The following parameters completely characterize the sensor *i*, where: $(x_i, y_i)$ coordinates of the sensor on the plane, $\theta$ - the maximum angle of coverage, $R_s$ - the maximum sensitivity range of the sensor, beyond which the target detection is not possible and the vector $\vec{d}_{i,j}$, which divides the coverage vector.

If the object *k*, located in $\vec{t}_k$ and is directed sensor *i* located in $\vec{l}_i$, then object presence in a coverage zone can be defined by an algorithm which is called presence in target sector test, Target In Sector (TIS) [76].

For this it is necessary to find a distance vector $\vec{v}_{i,k}$, defining the distance from the sensor *i* to the object *k*.

$$\vec{v}_{i,k} = \vec{t}_k - \vec{l}_i \tag{1}$$

Next, it is necessary to determine whether the distance vector is covered by sensor *i*, using the following test:

$$\vec{d}_{i,j}^T \vec{v}_{i,k} \geq \left\| \vec{v}_{i,k} \right\|_2 \cos\left(\frac{\theta}{2}\right) \tag{2}$$

Then, check whether the sensor can detect the object:

$$\left\| \vec{v}_{i,k} \right\|_2 \leq R_s \tag{3}$$

In case when conditions 2 and 3 are met, the test result is true, and the sensor *i* covers the object *k*.

### 1.5.2 Voronoi diagram

The approach is based on the Voronoi tessellation, which is widely used to determine the coverage in isotropic sensor networks. The idea is that to introduce a model of the sensor in the form of an ellipse with a certain orientation (Fig. 3b), to simplify the construction and analysis of the chart, eventually reducing the problem to the isotropic model (Fig. 3a) [76].

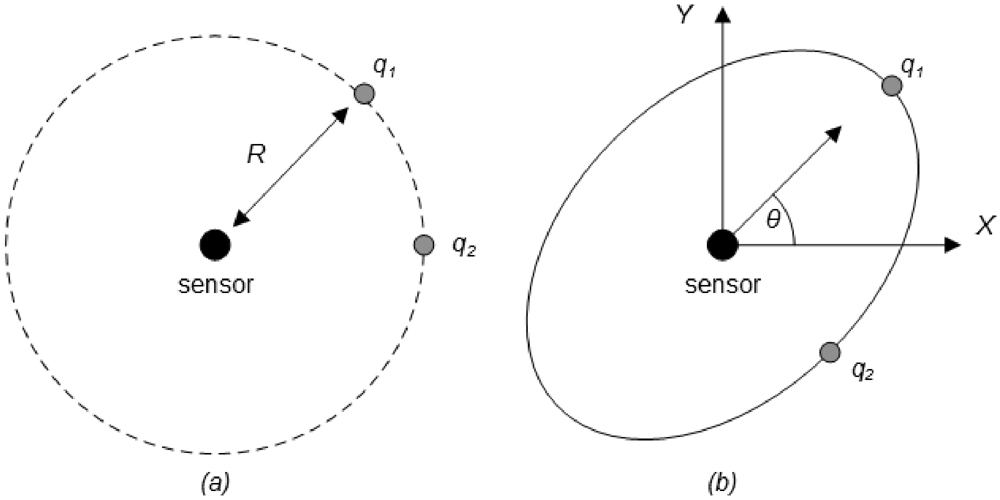



***Fig. 3.*** *Isotropic model (a), Anisotropic model (b)*

Anisotropic Voronoi region for sensor *i* in this case is defined as:

$$V_i^* = \left\{ q \in Q \,|\, \|q - p_i\|_{L_i} \leq \|q - p_j\|_{L_j}, \forall j \neq i \right\} \tag{4}$$

Where the point *q* belongs to the set of points in the region *Q*, and the distance expressed in non-Euclidean metric:

$$\|q - p_i\|_{L_i}^2 = (q - p_i)^T L_i (q - p_i) \tag{5}$$

Where the matrix $L_i$ can be expanded as $L_i = F_i^T F_i$, where

$$F_i = \left[ \begin{pmatrix} \frac{c}{a} & 0 \\ 0 & \frac{c}{b} \end{pmatrix} \begin{pmatrix} cos\theta_i & sin\theta_i \\ -sin\theta_i & cos\theta_i \end{pmatrix} \right] \tag{6}$$

Where $\theta_i$ angle (orientation) *i*-sensor, and *(a, b, c)> 0*, the ellipse parameters.

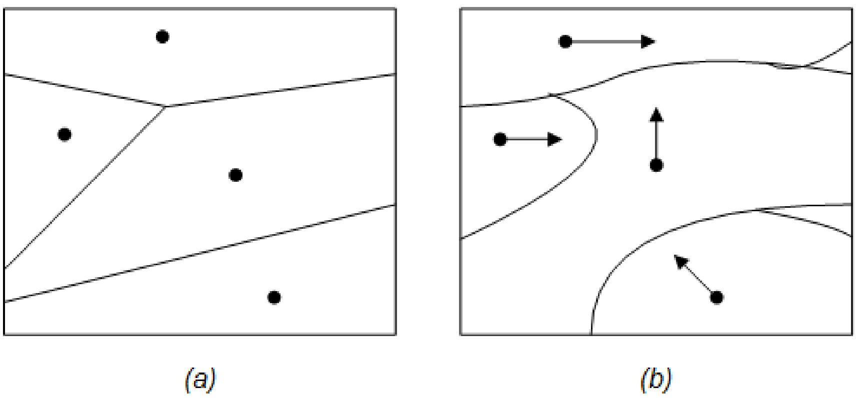



***Fig. 4.*** *Isotropic Voronoi regions (a), Anisotropic Voronoi regions (b)*

In this case, the Voronoi regions of the polygons becomes figures described by the second order curves which leads to discontinuities in the regions and occurrence of regions without generators and with undetermined coverage (Fig. 4b).

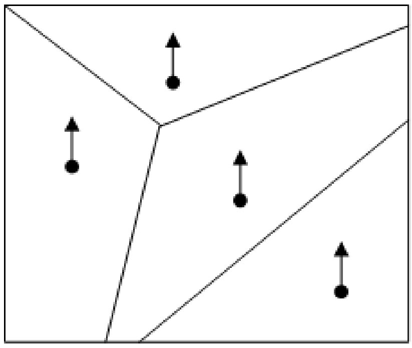



***Fig. 5.*** *Anisotropic Voronoi regions with the same orientation*

In order to bring the chart to normal view it is required to make another assumption - that all sensors are oriented in the same direction (Fig. 5). This substantially reduces the legitimacy of the method.

### 1.5.3 Probabilistic method

This approach does not determine the coverage area for each sensor, as is done in the Voronoi diagram. Instead, the calculated probability of object detection by the sensor. Moreover, several sensors total probability may be taken into account and, in case when the object is in the multiple sensors coverage area simultaneously [77].

The model depends not only on distance but also on the orientation of the sensor relative to the region of interest. Suppose each sensor has a limited coverage area $Q_i$ with a maximum range $R$ and the direction $\Theta$ (Fig. 6). The possibility of covering each sensor is limited by radial distance and angle relative to the object that should fall into the coverage area.

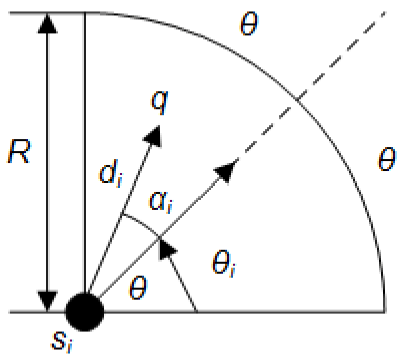



***Fig. 6.*** *Probabilistic coverage estimation*

Mathematically, the sensor $i$ characteristics are dependent on the distance $d_i$, orientation $a_i$ of sensor $i$ with relatively to the target $q$. The sensor coverage area defines as follows:

$$Q_i = \{q \in Q | d_i \leq R \wedge |\alpha_i| \leq \Theta\} \qquad (7)$$

Where:

$$d_i = \|q - s_i\|$$
$$\alpha_i = cos^{-1}\left(\frac{(q - s_i)(cos\theta_i \; sin\theta_i)}{\|q - s_i\|}\right) \tag{8}$$
$$\Theta \in \left(0, \frac{\pi}{2}\right)$$

It is also necessary to make an assumption that the sensor *i* has coverage only in its region $Q_i$.

$$p_i(q) = 0, \frac{\partial p_i(q)}{\partial d_i(q)} = 0, \frac{\partial p_i(q)}{\partial \alpha_i(q)} = 0 \; if \; q \notin \; Q_i \tag{9}$$

Thus, the probabilistic model of the sensor is:

$$p_i(q) = \begin{cases} \frac{(d_i - R)^2(\alpha_i - \Theta)^2}{R^2\Theta^2} & if \; q \in \; Q_i \\ 0 & otherwise \end{cases} \tag{10}$$

As can be seen from the review, the geometric estimating method has serious shortcomings that limits its use. First, the need of detected object approximate coordinate knowledge in advance. Also, it is impossible to determine the probability of the object detection, since the method gives only a binary estimation - in the zone or outside the zone.

The method based on the Voronoi diagram, though it is one of the most effective for networks with isotropic sensors, involves not quite realistic assumptions for anisotropic sensors. The elliptical shape of the coverage is not characteristic for the sensors used in the fields considered in this paper, as well as the probability of the same orientation of the sensors is very low. Without assumption of the same orientated sensors, this diagram cannot be used anymore, and this reduces its value.

The most promising is the probabilistic approach, as it makes possible to take into account coverage probability not only for each individual sensor, but also the total probability in case of overlapping zones. This method can be scaled to three-dimensional space. Applicability, algorithms and effectiveness of this method for coverage estimation considered typical for ITS scenarios may be the subject of further research.

### 1.6 Localization error

In the ideal case, localization goal is to determine coordinates of the node of interest with zero error components. Obviously, such a goal is unachievable due to a number of limitations inherent in the nature of the localization problem. First of all, the location will always be tied to a certain frame of reference, in the simplest case, the set of reference points, the coordinates of which in turn will contain measurement errors. Another source of error is inaccurate measuring instruments [72].

For most measurements carried out using sensors, regardless of whether such measurements are related localization, relative distance measurements or other types of measurements, characterized by binding to a precise time of measurement for the correct comparison to the measurements made at other sensors.

Only in the case where the wireless network supports some time synchronization protocol, efficient measurement data sharing between adjacent nodes in the network becomes possible. It should be noted that the usually time synchronization protocols used in fixed networks do not provide sufficient stability and accuracy required for localization tasks.

A network node *i* may have a known or unknown location $p_t^i$, at each time *t*. Depending on the type of measurement and one, two or more nodes involved in the evaluation of the node location, the localization function is as follows:

$$x_t^i = h_{type}\left(p_t^i\right) + e_t^i \tag{11}$$

$$x_t^{ij} = h_{type}\left(p_t^i, p_t^j\right) + e_t^{ij} \tag{12}$$

$$x_t^{ijk} = h_{type}\left(p_t^i, p_t^j, p_t^k\right) + e_t^{ijk} \tag{13}$$

Where $x_t^i$ is a result of the evaluation, $e_t^i$ error component of this evaluation, and in turn $h_{type}$ indicates the type of the localization algorithm.

In all known models of measurement noise component $e_t$ is necessarily present. First and convenient approximation consists of assuming that the modified value consists of undistorted components and noise components that is a white or Gaussian components with a standard deviation $\sigma_e$. The corresponding measurement error depends on the measurement method and network architecture [62]. Thus, it is assumed that the measurement error is Gaussian with the probability density function:

$$P_E(e_t) = N(0, \sigma) \tag{14}$$

Achievable accuracy (Table 1.) of a rangefinder system is limited by following four primary factors: noise, time synchronization, sampling artifacts and effects of the multiple paths signal propagation. These factors contribute to the random measurement errors varying in space and time than reduces measurement accuracy. Since introduced errors are stochastic in nature, they cannot be fully compensated, although certain measurement techniques may reduce these errors at certain extent.

The noise and clutters introduces to the measurements random component of the error. Range measurement, distorted only by noise, is limited in its precision by a signal / noise ratio and bandwidth frequency devices.

***Table 1.*** *Mathematical notation of sensor observations in wireless sensor networks*

| Type | Measurements | Precision |
|---|---|---|
| Angle of arrival | $h_{DOA} = angle(p_t^i - p_t^j)$ | 5º - 10º |
| Time of arrival | $h_{TOA} = \lVert p_t^i - p_t^j \rVert$ | 5 – 100 m |
| Time difference of arrival | $h_{TDOA} = \lVert p_t^i - p_t^j \rVert - \lVert p_t^i - p_t^k \rVert$ | 10 – 60 m |
| Interferometric | $h_{TDOA} = \lVert p_t^A - p_t^D \rVert - \lVert p_t^B - p_t^C \rVert + \lVert p_t^B - p_t^C \rVert - \lVert p_t^A - p_t^C \rVert$ | 0.1 – 1 m |
| Received signal strengths | $h_{RSS,log} = P_0^i - n_{i,j}(\lVert p_t^i - p_t^j \rVert)$ <br> $h_{RSS,lin} = \overline{P}_0^i \lVert p_t^i - p_t^j \rVert^{-n_{i,j}}$ | 4 – 12 dB |
| Digital map information | $h_{MAP}^j(p_t^i, p_t^j)$ | (RSS MAP 3dB) |
| Position estimation Inertial sensors | $h_{POS} = p_t^i$ <br> $h_{INS}(p_t^i, p_t^j)$ | 5 – 20 m (GPS) |

Time synchronization in high precision measuring systems needed in the case where the distance estimated by the time difference between signals transmission and receive. For wireless networks, the synchronization accuracy usually is near 1 μs, what may lead to estimation error in range up to 300 meters. However, in high-precision and expensive systems, synchronization precision may reach 10 ns, what is equivalent to an error of 3 meters.

Errors resulting from the effect of the sampling signal, limit the accuracy of the size of $c/_{f_s^2}$, where $f_s$ - sampling frequency receiver, $c$ - the speed of light. The resulting measurement error has a uniform probability distribution and is expressed by the formula:

$$\sigma_{sample}^2 = \frac{1}{12} f_s^2 \tag{15}$$

Even in cases where a measuring system has the minimum error of the listed components, the system often can fail due to the fact that measurements are made not in completely open and free from obstacles environment. In a real life situation, the RF signal can be mirrored by foreign objects, resulting in the incoming signal receiving by several paths of different length.

## 1.7 LBS categories

To formulate the requirements for the localization algorithms and problems caused by them it makes sense to classify them dividing localization data consuming applications into four types: the applications of traffic safety, mobility, efficiency and telematics applications [78].

Sharing information through timely transmission of relevant data, taking into account the location of network members, enables resources optimization and creation of an intellectual environment for vehicles equipped with the appropriate hardware.

If for some applications is sufficient to determine the relative coordinates of the other vehicles, other applications require accurate absolute coordinates associated with the coordinates of digital map data which also contain some error. The interpretation of these features in the context of the above mentioned types of applications and requirements for accuracy of localization is shown in Fig. 7.

*Traffic safety:* applications that support traffic safety need the most accurate localization data, including association with digital maps. In this case, localization errors exceeding 5 meters are unacceptable. This is especially actual for lane identification problems, where the requirements for the error value are even higher and do not exceed 1 meter.

*Mobility:* accuracy requirements are less stringent, application allows localization error up to 8 meters. The problem is reduced to the identification of the road on which the vehicle resides, without specifying the lanes.

*Efficiency:* allowable limit of error is within 10 meters and can be provided by a large number of vehicles involved in the information exchange. Unlike safety applications where the number of participants is limited both temporally and spatially, in efficiency and mobility applications large number of users can enable alternative localization results verification methods.

*Telematics:* telematics applications mainly operate by zones and the problem reduces to the identification of areas in which the vehicle resides. Allowable error may be of the order of 15 meters and more. On the other hand, these applications often may handle large amounts of data.

Equipping of all VANET participants objects with global satellite systems receivers (GNSS) could be the solution to the localization problem. However, this solution requires a constant line of sight between the receiver and the satellites, and cannot provides localization in underground areas, indoors and in the presence of various obstacles. Thus, the problem of localization remains partially open. In addition, most used methods of cooperative localization are based on optimization techniques and assigns local minimum coordinates to non-localizable nodes, what is unacceptable for traffic safety applications. Possibility of not unique node local coordinates of placements in the global coordinate system problem also remains unsolved.

Another essential requirement for localization methods is to provide high availability and high reliability of the data. Ideally, the localization system should include several data sources, for example, GNSS, dead reckoning systems, various sensors and landmark recognition systems [79].

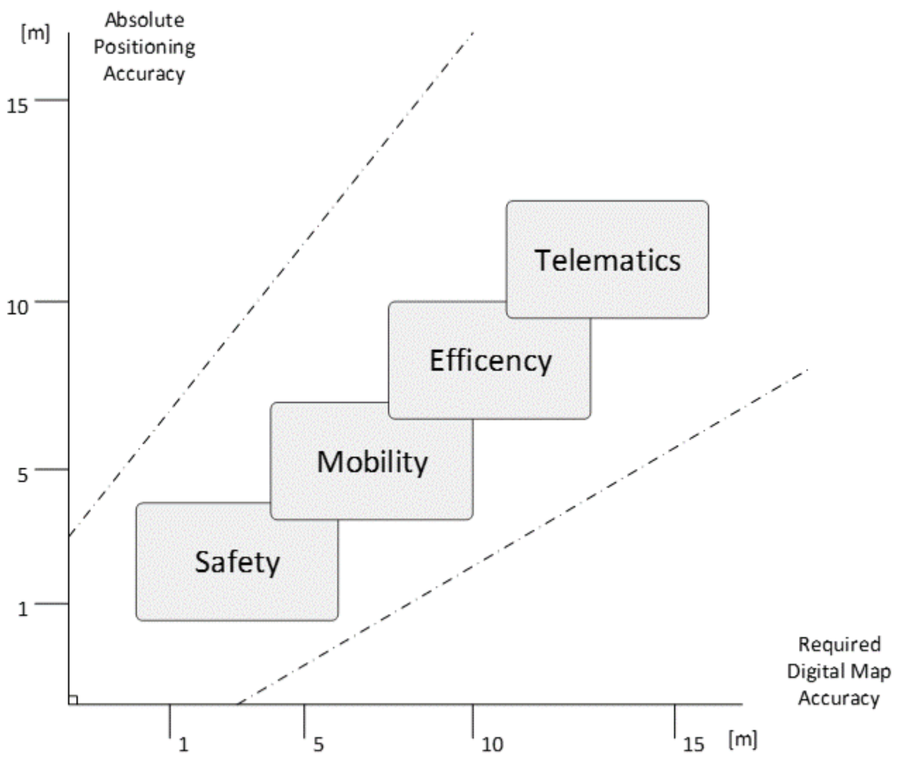



***Fig. 7.*** *Requirements for localization accuracy distribution by application domain*

To sum up the above, the requirements for localization methods can be characterized as follows:

- Accuracy: defined as the closeness of the position estimate of the vehicle to its actual location;
- Availability: the number of vehicle location estimates, which can be obtained per time unit;
- Response time: the time required by the localization procedure to acquire position estimate;
- Integrity: the level of trust that can be applied to the vehicle position estimation results.

In this chapter three important issues that arise in the development of Location Based Services in intelligent transportation systems were addressed: localization of wireless network objects, localizability and network coverage evaluation. Based on the above, we can conclude that, despite significant progress in the methods of localization, based on the combined use of different sensors and GPS, still there are situations in which these tools are ineffective or do not apply. It is obvious that the subject area needs improvements of existing and development of new localization methods.

# 2. LOCALIZATION METHODS

This chapter examines the factors that contribute to the difference in the approaches to the localization methods classification. Discussed the most common types of measurements on which the algorithms based localization. Provided an overview of the two main groups of localization methods, alternative to classical global positioning: methods without reference to the reference points designed to work in local coordinate systems and with conventional distances and methods with reference points designed to work with the actual topology and global coordinate systems. In conclusion, separately addressed cooperative localization in transport.

## 2.1 Localization taxonomy

Since each localization approach is designed to address different problems [50, 80, 81, 82, 83, 84, 85, 86 and 87] and support different applications [88, 89, 90, 91, 92, 93, 94, 95 and 96], they differ in many aspects, such as the measurement parameters used to determine the location, types of sensors, software and hardware requirements as well as the necessary infrastructure. So far, there is no common classification covering all aspects of localization methods, which can be applied to any technical solution. However, in order to provide a clearer picture and better assess the possibilities when selecting localization algorithms, further we propose a classification that largely takes into account and generalizes approaches mentioned in [97, 98, and 99].

Localization algorithms for wireless networks can be classified by the structure of the network, the conditions of applicability and limitations in the use of various algorithms.

### 2.1.1 Range free and range based methods

A significant part of the localization methods assumes a network node, with a precisely known location in a global coordinate system. Such a node called a beacon or an anchor, and such methods called techniques with a reference point. The essence of these methods can be formulated as the problem of finding a consistent set of nodes in the network locations given the available information about the relationship between nodes and between nodes and anchor.

Anchors in this case are the devices that may periodically transmit certain signals that can be used for localization, or sensors, that can obtain information about its location from other sources (e.g., via GPS). As with the anchor point methods involve presence of preset infrastructure in some specific application domains they may not be used.

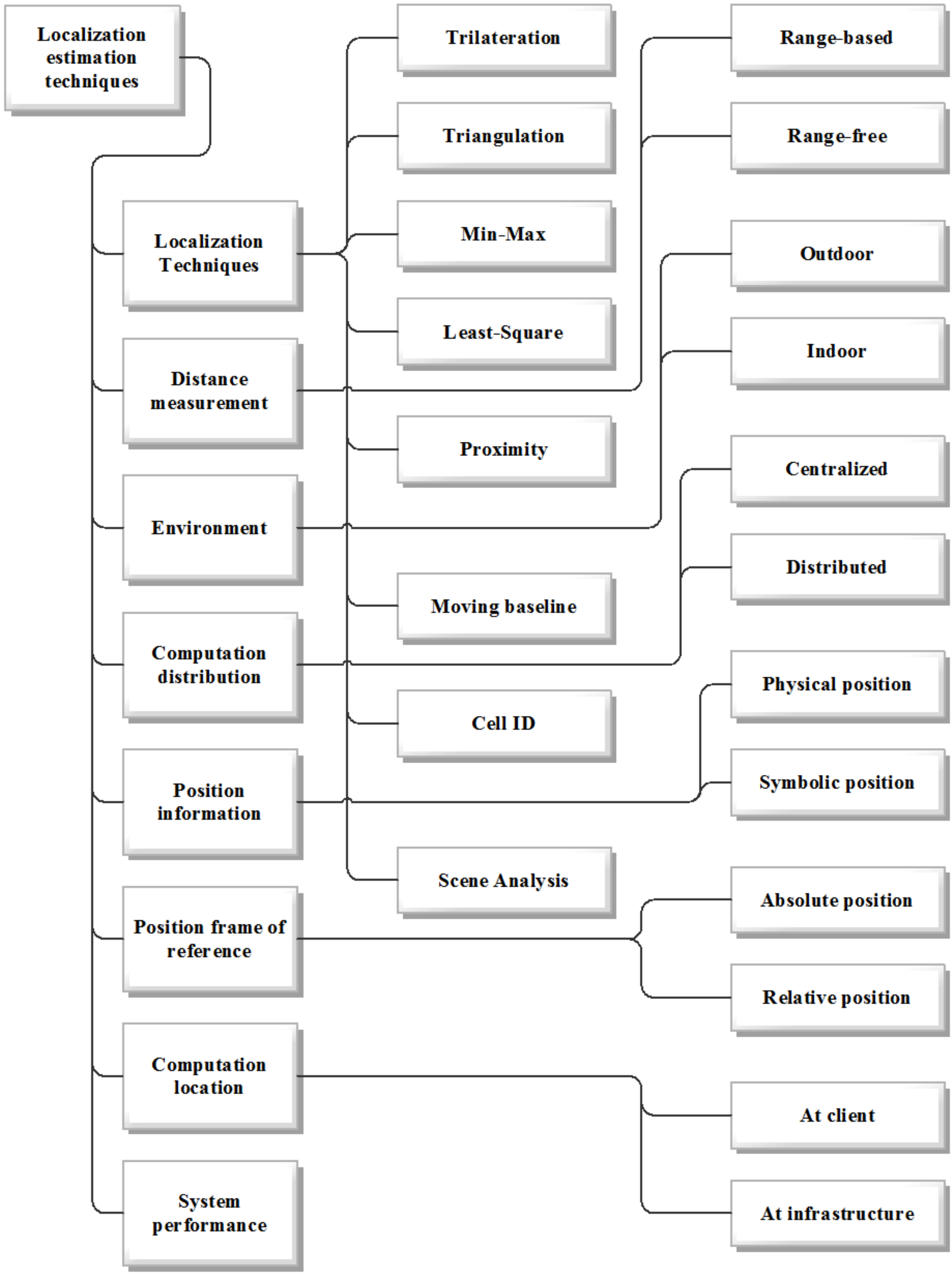



***Fig. 8.*** *Localization methods taxonomy*

On the other hand, methods without a reference point do not require specialized equipment or anchors to use data received from the outside of network. Instead of calculating the absolute nodes location, these methods allow to determine only the relative positions of the nodes in the local coordinate system with an arbitrary starting point. When a number of anchors in the network is sufficient (for two-dimensional space of at least three), the localization results obtained by methods without a reference point may be transformed into an absolute coordinate system.

### 2.1.2 Centralized and distributed localization methods

In the case of centralized algorithms, basic calculations are performed on a dedicated computer or a node located outside the network, while distributed algorithm computational load can be shared between all network nodes. This classification is directly connected with the formulation of the localization problem. If the problem is formulated as a global optimization, such as convex optimization [100], the node performing the calculations, should have all the data on the network distances and calculations must be performed centralized.

Implementation of centralized algorithms, usually leads to collisions and concurrence in the calculations during the necessary communications network with a central hub.

### 2.1.3 Proximity, distance and angle of arrival

To determine the location of the network node, localization algorithms can use data obtained from measurements of various types and of different nature. The most commonly used data types are the distance between nodes, the angle of arrival to the signal reception unit and the proximity factor characterizing the visibility or invisibility of node depending on its distance from a reference point.

The most accessible in terms of its obtaining is node proximity data, but it provides only a rough estimate of the location that can be improved only by a large number of measurements.

Another type of data used for localization is the measured distance between nodes, obtained by one of the methods such as the received signal strength (RSS), signal time of arrival (ToA), signal time difference of arrival (TDoA) or by some other means.

Scheme of the localization methods main taxonomic factors shown in Fig. 8. More detailed description of the factors given in Table 2.

Evaluation of distance using RSS is based on ratio of the received signal power and distance, where the dependence is modeled by the propagation loss of signal equation:

$$P_r = c\,\frac{P_t}{d^\alpha} \tag{16}$$

Where $P_r$ - received power, $P_r$ - transmit power, $c$ - the speed of light, $d$ - distance and propagation loss exponent signal $\alpha$, usually takes between 2 to 5 [101].

ToA method of measuring the distance uses the relationship between distance and signal propagation time. If the transmitter and receiver are well synchronized, the distance can be calculated using the time stamp included in the messages.

***Table 2.*** *Localization methods taxonomy description*

| # | Name | Description |
|---|---|---|
| **1** | Outdoors | Metropolitan environment; obstacles (if any) of large dimensions |
| **2** | Indoors | Obstacle-based environment |
| **3** | Range based | Absolute positions into the system are computed from absolute distance measurements; prone to cell graning and symbolic location |
| **4** | Range free | Relative positions are computed in a large network, to anchor nodes, or in terms of hop counts |
| **5** | Triangulation, Trilateration, Multilateration, Min-Max, Least-squares, Proximity, Scene analysis, etc. | Location estimation technique |
| **6** | Centralized | Location is determined at a single site, be it a client or an infrastructure |
| **7** | Distributed | Location is inferred from aggregated input at various sites |
| **8** | Physical positioning | Coordinates relative to a large, flat coordinate system |
| **9** | Symbolical positioning | Abstract positioning, relative to artificial reference points |
| **10** | Absolute positioning | The same location grid of reference is used for all located objects; all locations can be compared |
| **11** | Relative positioning | Each object is located within an own reference grid |
| **12** | Client based | For the client to computer its own location, the network emits telemetry |
| **13** | Infrastructure based | For the infrastructure to compute the object's locations, the object must emit telemetry |
| **14** | Accuracy | Grain size, in units of distance, reached by a certain set of computation units |
| **15** | Precision | Frequency (probability) of making a computation of a certain accuracy |
| **16** | Portability | The possibility of porting a location system to different sites without heavyweight calibration |
| **17** | Self-organization | The non-reliance on fixed infrastructure and ad-hoc fashion adaptation to random site conditions |
| **18** | Cost/Power consumption | Cost of hardware, deployment, maintenance; power consumption, if the case |
| **19** | Scalability | Client load per unit of physical infrastructure and the infrastructure expansion limit |
| **20** | Security | Information access policy |

The latter considered type of measurement - the angle of arrival at reception signal node (AoA), applicable when obtaining the necessary data by using a directional antenna or antenna array. Since the AoA method requires multiple antennas, as well as limited by the size of devices and their complexity, it is less suited for use in wireless sensor networks. However, in cases when such data is available, the network can be localized using triangulation.

### 2.1.4 Static and dynamic networks

The majority of modern sensor networks localization algorithms do not provide explicit functionality for the mobile networks, assuming the fact that the network is a static object. In reality, however, an increasing number of applications are working with dynamic networks that actively changes their topology. Thus, modern localization algorithms must take into account these characteristics of wireless networks.

### 2.2 Global positioning

The basis of determining the coordinates of the GPS-receiver is calculating the distance from it to several satellites, the location of which is assumed to be known [102, 103]. Known distance from the receiver to three satellites theoretically is enough to determine the coordinates of the GPS-receiver based navigator. In practice there is always some error in the measurements, e.g., due to inaccurate synchronization of the receiver clock and satellite, influence of the atmospheric condition etc. Therefore, to determine three-dimensional coordinates of the GPS-receiver uses at least four satellites. Receiving the signal from four (or more) satellites, GPS-receiver searches for the corresponding point of intersection of the spheres. If no such point, the GPS-receiver processor begins by successive approximations to adjust its data until achieves all spheres intersection at one point.

The coordinates of the mobile unit are determined using standard GPS-navigation receiver built into the user terminal. Navigation receiver for GPS system consists of a receiving unit and a small sized antenna with low noise amplifier. The receiving module is available either as a standalone unit with integrated power supply or as a separate card inserted in the subscriber terminal.

The device usually uses its own miniature antenna and autonomously calculates the geographic coordinates and universal time (UTC) by the navigation signals. GPS-receivers usually allows determining their own position with an accuracy of less than 100 m. After the signal acquisition, navigation receiver automatically calculates the coordinates of the object, the speed, universal time, and generates a report. Information about the location of the object transmitted via satellite to the control destination point. Navigation devices can vary in number of the reception channels, update rate, computation time, accuracy and reliability of the coordinate determination.

Modern GPS-enabled devices usually equipped with six to eight receivers that allows tracking virtually all navigation satellites that are within range of the object [104]. If the channels number is less than the observed satellites, device automatically determines and selects the optimal combination of satellites. Navigation data update rate usually is 1 second.

The detection time depends on the number of simultaneously observed satellites and positioning mode. Navigation parameters are determined in two modes 2D and 3D. In 2D mode, only latitude and longitude is set (height is assumed to be known). It is sufficient presence in the three satellites in the line of sight. 2D mode the coordinate determining time is typically less than two minutes. To determine the 3D coordinates requires at least four satellites in corresponding area. Guaranteed detection time should be no more than 3-4 minutes [105, 106].

### 2.3 Range-free localization

Depending on the object, carrying out measurement and location calculation, range free location methods can be divided into three main types:

- Mobile device positioning:
  Device measures all the necessary data, approximates and calculates its own location relying only on this data;
- Network positioning:
  Network measures all necessary parameters for localization, approximates and calculates the location of the device while the device remains passive throughout the process;
- Hybrid positioning:
  Location calculation operation is distributed between the device and the network. As a rule, the device collects the necessary data for the algorithm, and the network, having a large computing power, makes the necessary calculations.

  Next, let us consider some of the most common range free localization algorithms.

#### 2.3.1 Sum-Dist

This method is described in [107] and is the most simple solution for estimating the distance from the selected node to anchor node. It consists of adding the distances of all the transitions between nodes during network traversal. Each node sends a message, which includes a unique node ID, the local coordinates and the length of the path, which is initiated by zero in the first round. When a node receives a message, it adds the distance to the transmitting node to the length of the path and sends a message further.

Thus, considering only the shortest paths, each node obtains estimated distance to the anchor node. For example, Fig. 9 illustrates estimated distance between nodes $S$ and $D$ as $d_{SY} + d_{YD}$, whereas $d_{SD} \geq d_{SY} + d_{YD}$ due to the triangle inequality. Let $x_1, x_2, \dots, x_q$ is a path from node $x_1 \in V \backslash \Delta$ to node $a \in \Delta$. Evaluation of distance defined recursively:

$$\hat{d}_{x1a} = d_{x1x2} + \hat{d}_{x2a} \tag{17}$$

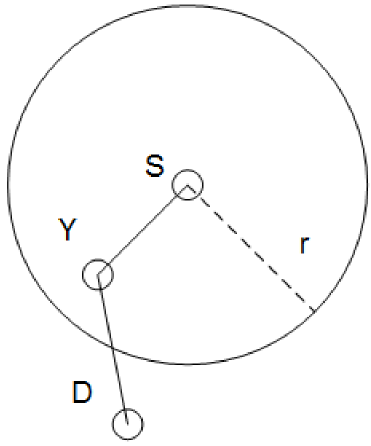



***Fig. 9.*** *Sum-dist method*

The advantage of this method is that it is extremely simple, and the algorithm is very fast. Moreover, the sheer number of computations is very small. On the other hand, there is an obvious drawback, which consists in the accumulation of errors in the distance estimation on each transition and distribution of errors over the network [108].

### 2.3.2 DV-Hop

Sum-dist algorithm drawback is that it collects and distributes the error as network traversal. In large networks, as well as in networks with a large distance between the nodes, these errors can reach a considerable value. Instead of summing distances containing errors, more robust alternative provides method based on topological information for counting the number of transit passages. This approach is called DV-hop and described in [109], as well as the Hop-TERRAIN in [110].

Essentially, DV-hop consists of two waves of network traversal. After the first wave, and in Sum-dist algorithm, nodes receive data about its location and the minimum number of hops to the anchor nodes. The second wave gauge traversal uses known transitions number that allows calculating new adjusted node location. Refinement consists of multiplying the number of hops to the average length of the transition. Once the node specifies its location, it passes the calibration data to the following nodes [111].

### 2.3.3 Network positioning

The advantage of network positioning for the mobile wireless devices is that it requires no additional hardware for location measuring and calculating. Need for of such additional hardware significantly increases cost of portable wireless devices. The most common example of network positioning is Cell-ID method, which idea is to use the GSM cells with known geographic coordinates to calculate positions [112].

The great advantage of Cell-ID method is the fact that the necessary infrastructure for positioning exists almost everywhere in the world. Each GSM station has its own unique ID/ Mac address. Thus, knowing the mobile station ID, it is possible to estimate the approximate location of the user.

However, since the GSM cells may have a very large coverage area, localization results may also contain a significant error component. In general, the accuracy Cell-ID method depends on the density of the mobile network in the area. In suburban areas, covering cell size can be up to 30 kilometers, while in the cities, the coverage may be considerably less, up to10 meters [112].

## 2.4 Range-based localization

Virtually all existing today, numerous range-based techniques of localization are based on the below listed basic methods and algorithms:

### 2.4.1 Triangulation

Triangulation is the process of positioning, based on angular measurements relative to a known reference points. Historically, the triangulation is most often used in geodesy, as in this application area angles are easily and accurately measured by simple methods. Applications of this method in the context of wireless localization systems are considered in detail in articles on Ad hoc positioning system (APS) using AOA and Error characteristics of ad hoc positioning systems (APS) [113, 114] .

Having data on coordinates of the triangle vertices and the angles at which the internal point “observes” the top, we can determine position of the internal point; internal point knows no distance to the vertices, and only the angles to the sides of the triangle (Figure 10).

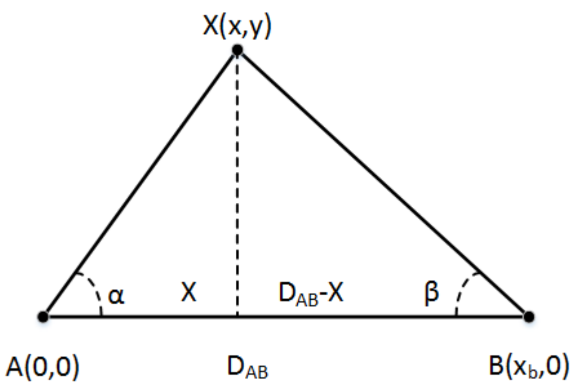



**Fig. 10.** Two-dimensional triangulation

$$x = \frac{\tan\beta}{\tan\alpha + \tan\beta}$$
$$y = \frac{\tan\beta \cdot \tan\alpha}{\tan\alpha + \tan\beta} \tag{18}$$

Where: *(x, y)* coordinates of localizable point, *α*, *β* opposing corners.

### 2.4.2 Trilateration

Trilateration also is one of the most widely used positioning methods, which implementations in the ad-hoc systems discussed in detail in Dynamic fine-grained localization method for ad-hoc networks of sensors [115]. Trilateration in two-dimensional space requires the measurement of distances up to three no collinear reference points, or four no collinear reference points in three-dimensional space. Two-dimensional example shown in Fig. 11.

$$\begin{cases} x^2 + y^2 = r_a^2 \\ (x - x_b)^2 + y^2 = r_b^2 \\ (x - x_c)^2 + (y - y_c)^2 = r_c^2 \end{cases}$$
$$x = \frac{x_b^2 + r_a^2 - r_b^2}{2x_b} \tag{19}$$
$$y = \frac{x_c^2 + y_c^2 - r_a^2 - r_c^2 - 2xx_c}{2y_c}$$

Where: *(x, y)* coordinates of localizable point, $r_i$ radius of the circle.

Additional empirical data can reduce the required number of measurements distances, if uncertainty of missing coordinates in the three-dimensional case can be resolved by other methods.

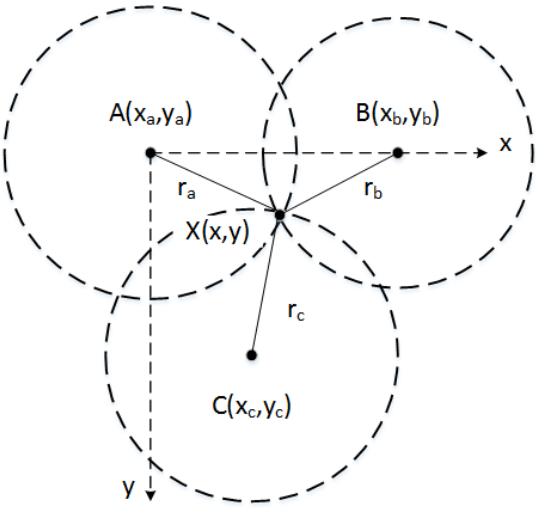



***Fig. 11.*** *Two-dimensional trilateration*

Three-dimensional trilateration problem (Fig. 12) is extended to the finding coordinates of intersection of three spheres, which are determined by solving the system of equations. To simplify the calculations, assume that the centers of all three areas lie in the plane *z = 0*, one of them coincides with the origin, the second - lies on the axis *x*. Imposed restrictions do not reduce the generality to this kind and any system equations can be brought to such form by moving to a another coordinate system. To find a solution for the original coordinate system, transformations are applied to the solutions found in this one (reduced) coordinate system, opposite to those that allowed the original set of three points align with appropriate constraints.

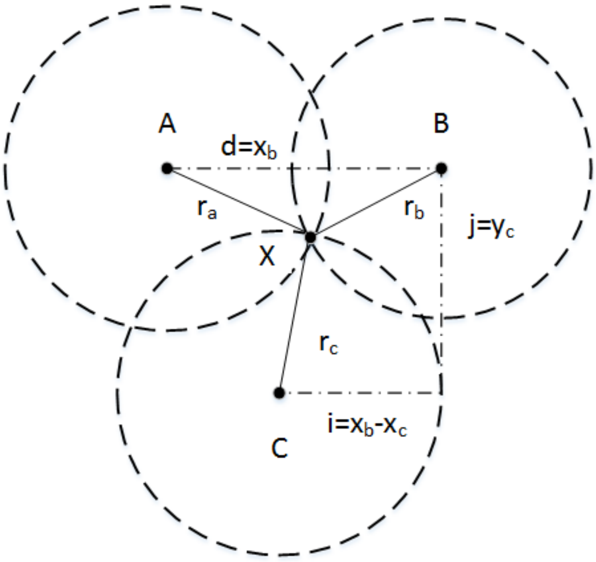



***Fig. 12.*** *Three-dimensional trilateration*

$$\begin{cases} x^2 + y^2 + z^2 = r_a^2 \\ (x-d)^2 + y^2 + z^2 = r_b^2 \\ (x-i)^2 + (y-y_c)^2 = r_c^2 \end{cases}$$

$$x = \frac{d^2 + r_a^2 - r_b^2}{2d} \tag{20}$$

$$y = \frac{r_a^2 - r_c^2 + (x-j)^2}{2j} + \frac{j}{2} - \frac{(r_a^2 - r_c^2 + d^2)^2}{8d^2 j}$$

$$z = \sqrt{r_a^2 - x^2 + y^2}$$

### 2.4.3 Multilateration

Multilateration is a navigation technique based on the measurement of the difference in distance to two or more reference points, which transmit signals at certain points in time. In contrast to the measurement of absolute distances and angles, the difference in the distance measurement can use arbitrarily large number of control points required for the desired accuracy.

The difference of signal arrival time τ from *n* reference points to the reference point *n + 1* makes it possible to calculate the coordinates in *n* dimensions (Fig. 13). Unlike trilateration, multilateration is atomic and requires the interaction of all participants.

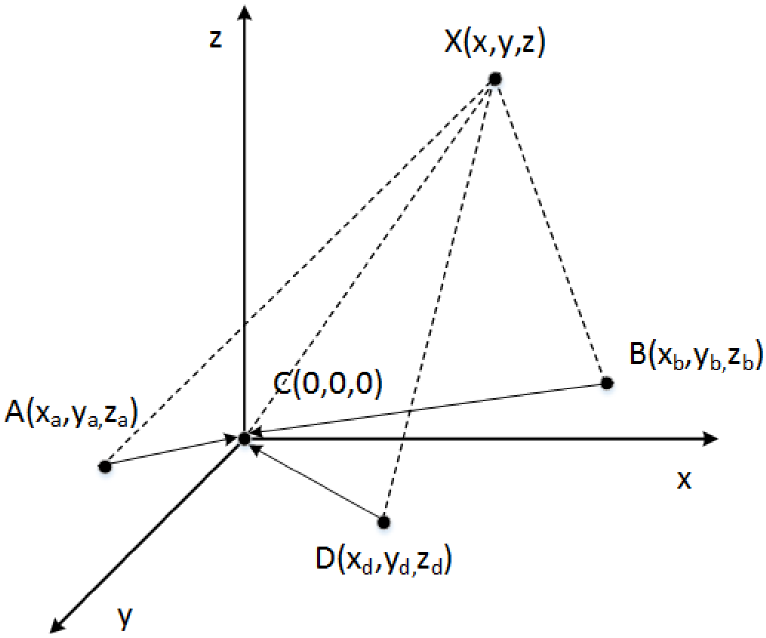



***Fig. 13.*** *Multilateration*

$$\begin{aligned}
\tau_a &= \frac{1}{c}\left(\sqrt{(x-x_a)^2+(y-y_a)^2+(z-z_a)^2}+D_{BC}-\sqrt{x^2+y^2+z^2}\right)\\
\tau_b &= \frac{1}{c}\left(\sqrt{(x-x_b)^2+(y-y_b)^2+(z-z_b)^2}+D_{BC}-\sqrt{x^2+y^2+z^2}\right)\\
\tau_c &= \frac{1}{c}\left(\sqrt{(x-x_c)^2+(y-y_c)^2+(z-z_c)^2}+D_{DC}-\sqrt{x^2+y^2+z^2}\right)
\end{aligned} \tag{21}$$

Where: $\tau$ time difference of signal arrival, *c* speed of light.

### 2.4.4 Min-Max

Since laterations can be computationally complex and may contain measurement error, in some cases, a simplified method is applied where the position is taken at the intersection of the diagonals of the rectangle formed by the intersection of minimal bounding squares circles, centered at the reference points and a radius equal to the measured distance (Fig. 14).

$$[max(x_i-r_i), max(y_i-r_i)] \times [min(x_i-r_i), min(y_i-r_i)] \tag{22}$$

Wherein: *(x, y)* coordinates of localizable point, $r_i$ radius of the circle.

The possibilities of using this method in localization systems are considered by in Robust Min-Max Localization techniques proposed in [116].

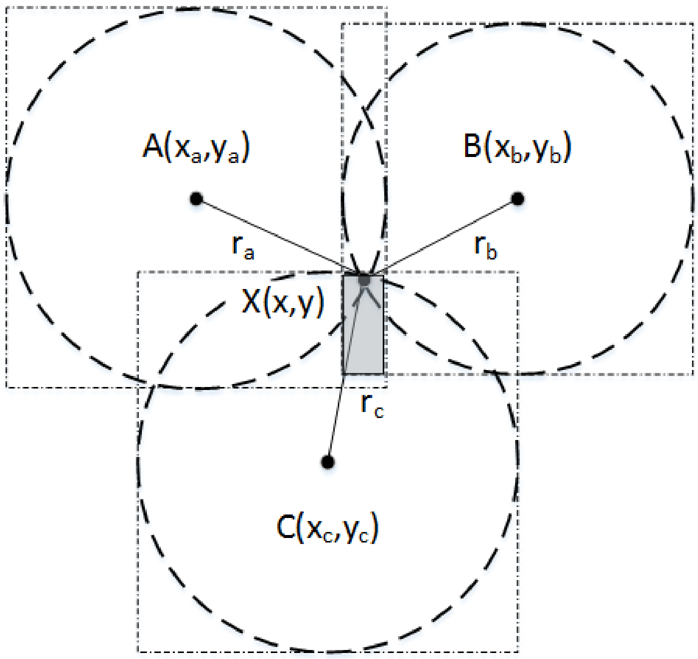



***Fig. 14.*** *Min-Max*

### 2.4.5 Clustering localization

A distinctive feature of the cluster localization method proposed in [117], is that it remains functional even with a significant component of the error in the measurements. The principle of the algorithm is as follows:

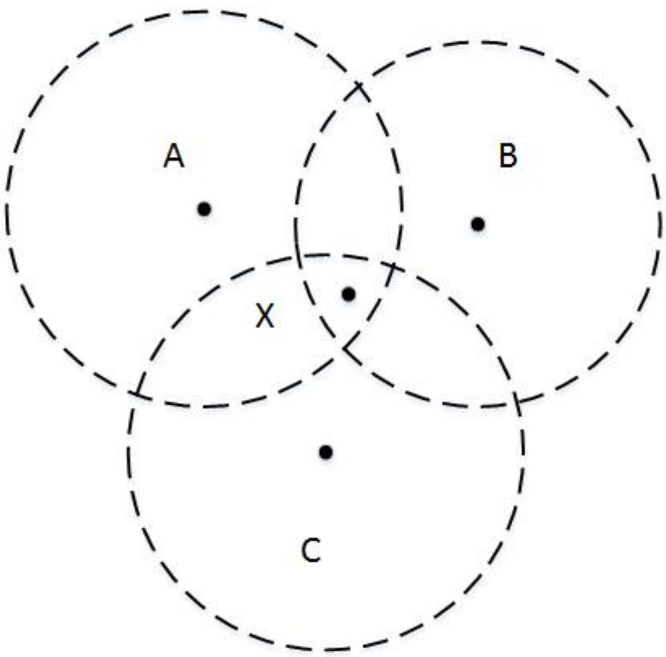



***Fig. 15.*** *Clustering localization*

The measured distance from the reference points *A*, *B*, *C*, or more to localized point *X*, forming a circles of radiuses $r_a$, $r_b$ and $r_c$. Since the coordinates of the centers and obtained radii of the circles contain errors, the circles will not have a single point of intersection, but will have some region where they overlap, as shown in Fig. 15. Point *X* with unknown coordinates will be within this region. Thus, for the three circles there are six points of intersection. Three of them are close one to another, forming a cluster. The point *X* is inside the cluster.

To find the coordinates of point *X* it is necessary using the clustering method to calculate the distance between each pair of points of intersection of the circles. Then, select of them two points with the smallest distance as the basis for the cluster. Calculate the centroid of the cluster. Then select the next point of intersection of the circles with the smallest distance to the centroid obtained. Then, include this point in the cluster and then calculate its centroid. Repeat procedure up until the size of the cluster reaches k, where k - number of circles. Centroid of the resulting cluster corresponds to the coordinates of *X* [118].

### 2.4.6 Least squares method

Having information about the distances from the object *X* at unknown location to three reference points *A*, *B* and *C* with known coordinates it is possible to set up a system of equations defining the relationship between the coordinates *X*, reference points coordinates, and the distance to them.

$$\begin{aligned}(x_x - a_x)^2 + \left(x_y - a_y\right)^2 - a_r^2 &= 0\\ (x_x - b_x)^2 + \left(x_y - b_y\right)^2 - b_r^2 &= 0\\ (x_x - c_x)^2 + \left(x_y - c_y\right)^2 - c_r^2 &= 0\end{aligned} \tag{23}$$

The resulting system of three or more equations with two unknowns: $\left(x_x, x_y\right)$. Since the number of unknowns exceeds the number of equations, the system is over defined and generally has no unique solution. Nevertheless, it has the least square solution. Replace the zeros on the right side with non-zero residue values.

$$\begin{aligned}(x_x - a_x)^2 + \left(x_y - a_y\right)^2 - a_r^2 &= a_\Delta^2\\ (x_x - b_x)^2 + \left(x_y - b_y\right)^2 - b_r^2 &= b_\Delta^2\\ (x_x - c_x)^2 + \left(x_y - c_y\right)^2 - c_r^2 &= c_\Delta^2\end{aligned} \tag{24}$$

Least squares solution is a unique solution $\left(x_x, x_y\right)$, which minimizes the sum of squared residuals $(a_\Delta^2 + b_\Delta^2 + c_\Delta^2 + \cdots)$. The system of equations is nonlinear in the parameters of the solution $\left(x_x, x_y\right)$ and to obtain the solution it requires the use of a non-linear least squares algorithm. The algorithm starts with the initial assumptions about the decision and held a number of repetitions. Each repetition corrects solution so that the residual sum of squares is reduced. The algorithm may also take into account information about the probabilistic characteristics of the input parameters. More details of such algorithms are considered in the literature on numerical methods [119].

### 2.4.7 Proximity factor

Localization method based on the proximity factor [120], can be formulated in terms of graph theory [121]. A wireless network is represented as a graph *G(V, E)*. Subset of nodes *H⊂V*, know their own location $(p_1, p_2, p_3 \dots p_m)$. Proximity factor assessment usually is interpreted in two models:

- Adjacency matrix
- Distance Matrix

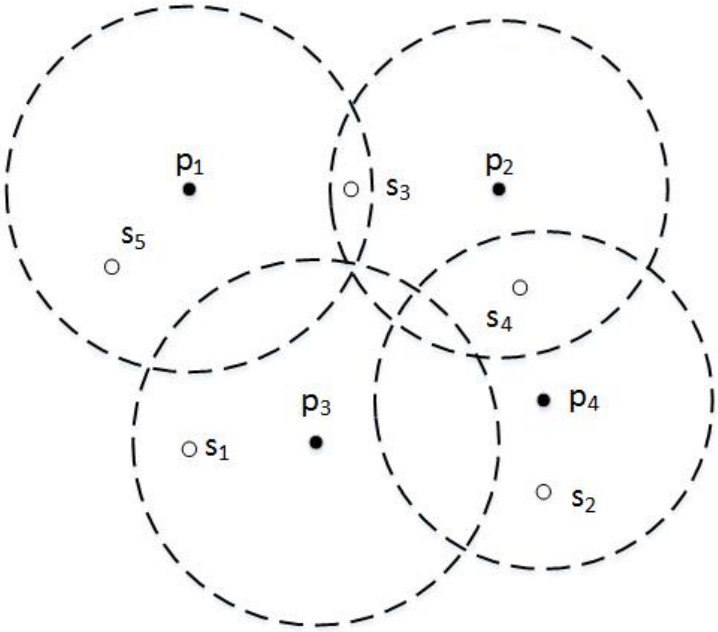



***Fig. 16.*** *Proximity factor localization example*

The task is to estimate locations $(s_1, s_2, s_3 \dots s_{n-m})$ of the V-H set of other network nodes. Example localization of sensors having information about their proximity is shown in Fig. 16. Various sensors can have a different number of reference points in its coverage area, and the location estimation accuracy increases as a function of the number of reference points available for sensors.

### 2.4.8 Moving baseline

This localization method of wireless networks, as well as many others, is based on the fact that the network nodes are devices capable of measuring distances and to communicate with a subset of other network nodes. Method, based on these data, suggests to reconstruct globally consistent network topology, however, unlike the previous methods, implies that the nodes are in motion. Thus, there is no external data and predefined reference coordinates on preset infrastructure.

The method was developed on the basis of actual localization scenarios where a group of people or robots must perform certain tasks under the unavailability of GPS signals. The collection of such methods is called a moving baseline. A necessary condition is the ability of each moving node to generate a series of range measurements to the rest of the network and provide them with a time stamp. Also, it is assumed that each device has and can tell its own unique identification number to other devices within its coverage area.

The choice of these system parameters is made due to the capabilities of existing devices such as Crickets [122] and UWB (ultra-wideband) radio [123]. This method solves the problem of localization by range measurements Time Series associated with a particular node ID, and further, comparing these measurements, what allows to reconstruct the global picture of all node movements.

Reconstructed nodes paths, using hyperbolic approximation, trilateration, subgraphs alignment and turning points recognition allow to solve these challenges and to evaluate the location of moving network objects [124].

### 2.4.9 Scene analysis

Localization method based on scene analysis suggests the possibility of monitoring some space around the object being localized. Typically, this is done using a set of sensors covering the required space and capable of perceiving visual, infrared, or other data. Of particular note is the localization techniques with advanced collection of identification labels on the ground.

Methods with identification marks mainly consist of two stages [125]. The first step is to gather information about the environment, which localization will be performed, and development of an environment identification labels database. The most common tags used for data rates are received signal strength RSS (Received Signal Strength). The two most common methods of this type are the method of k-nearest neighbor (kNN), also known as the radio map, and the probabilistic method.

KNN method consists of an initial measurement of RSS in certain reference points and building RSS database, also known as radio map. Further, in the process of localization of the object, observed RSS measurements are used to search the database for *k* most similar measurements, whereupon the location of the object relative to the labels found estimated by least squares error.

Probabilistic approach to the problem is formulated as finding the location of the object from *n* options, given RSS observation vector, using Bayesian a posterior probability formula. Thus, the location is selected with the highest probability of the object. Generally, a probabilistic approach includes several steps, such as data calibration, training, errors estimation, and the accumulation of historical data.

Furthermore, there are methods based on various combinations of scene analysis, proximity factor, such as the location method for Bluetooth ad-hoc described in [51]. Bluetooth devices periodically send requests in the form of sequences of signals of different power levels to discover new nodes in the network. Request for low power devices detected in close proximity, high power requests discovers more remote device that allows one to get a first rough estimate of the distance. Nodes exchanges produced observations data and the presence of a network anchor node allows tying this information to a common coordinate system. The error in results of localization method described may not exceed 1.88 meters.

### 2.5 Cooperative localization

Currently, a large number of studies in the field of ITS is directed to cooperative localization systems with V2V and V2I communications. Numerous projects at the European and national level focus on this topic in order to improve the safety and efficiency of traffic [126].

Such systems usually rely on the exchange of pseudo ranges obtained using GNSS, between several vehicles available for them over the communication channel. Assuming that in scope of vehicles are the same navigation satellites, occurring systematic errors can be partially compensated, and the relative locations vector can be determined with sufficient accuracy. After performing the necessary calculations, all the information obtained (GNSS data, vector relative locations, the data on speed, acceleration, etc.) can be used for further processing algorithms with high-tech and complex data filters.

Speaking about unsolved problems inherent in cooperative positioning systems, it should be noted that the information distributed in the network, is not independent and therefore a data correlation may occur. This can lead to the appearance of erroneous and contradictory data, if such correlation errors will not be compensated in one or another way.

This chapter described the classification of localization methods and the types of measurements required for the implementation of the algorithms. The algorithms that uses as input conditional distance determined by the number of transit transitions between nodes. In addition, considered algorithms that use data on proximity factors to the nodes reference points, as well as algorithms based on clustering methods, storage and analysis of measurements time series.

In addition, greater precision algorithms were considered, requiring a measure of Euclidean distances, angles, time, signal propagation delay and other physical parameters of the environment based on optimization techniques, numerical methods and the methods of computational geometry. Specifically mentioned the problem of dependence and measurement correlation errors inherent in the use of cooperative localization systems in transport.

# 3. ANALYSIS OF RELATIVE POSITION AND CHARACTERISTICS OF REFERENCE POINTS ON THE OBJECT LOCALIZATION ERROR IN VANET NETWORKS

An important part of the localization problem is to provide the most accurate origin estimation of the localized object in a difficult measurement conditions, noise, distortion or absence of the signal line of sight. Other factors affecting the accuracy of localization are reference point location scenarios used for coordinates calculations, as well as the parameters of the probability distribution and the nature of errors, both in the measured ranges and errors resulting localization with inaccurate data.

This chapter discusses in detail and compares experimentally localization techniques such as least-squares method using a Levenberg-Markarth optimization algorithm [127], cluster method and Min-Max algorithm. Investigates the effects of different error distributions and their combinations on the localization results. Experimentally obtained data showing the effect of different scenarios in resulting error distribution in the object localized coordinates, including scenarios where localization becomes either extremely difficult or not possible at all.

## 3.1 Graph vertices coordinates error characteristics

Sources of error in the localization system can be divided into three main classes [128]: placement errors, channel errors and algorithmic errors. Placement errors associated with both the initial configuration of the network, its density, size, and inaccuracies in the anchor nodes placements with a priori known coordinates. Channel errors determined by the properties of the measured physical parameters that allow calculating the distance between nodes of the network, and by properties of the communication channel, used for measured data exchange. The magnitude of this error component is largely dependent on used technology and the environment changes in which localization tasks are performed. The overall assessment of the channel component of the measurement error is extremely difficult and must be done individually for specific technological solutions.

In the solutions using the principle of sharing coordinates where each newly localized node can become a new reference point, measurement error component, usually one way or another is accumulated, what in turn makes estimation task even more difficult.

To control and minimize the spread of errors [129] it is necessary to have a mechanism to evaluate the relative accuracy of the newly localized nodes and to choose from them the most reliable reference points.

Considering trilateration algorithm as an example in [130] proposed to introduce a metric of trilateration quality: Quality of Trilateration (QoT) based on the observation of trilateration accuracy, depending on relative geometrical arrangement of reference points.

QoT metric describes the differences in the placement of control points numerically and allows selecting the optimum combination from the possible. This mechanism also allows marking those placing of anchor points, which will have a significant uncertainty of result and a higher probability of reflection.

Trilateration quality *t* defined as: $Q(t) = \int_p f_t(p)dp, p \in Disk(p, R)$, where *t = Tri (s, {1,2,3})* means node s trilateration, based on three reference points $s_i$.

In this expression $f_t(p) = \prod_{i=1}^{3} f_{si,sj}\left(d\left(p, p(s_i)\right)\right)$, the probability density function for any point *p* in the plane, where *p(s)* means real node *s* location; $d\left(s_i, s_j\right)$ is the real distance between two adjacent nodes $s_i$ and $s_j$; $f_{si,sj}(x)$ probability distribution ($x \in [0, +\infty]$ - the distance value); expression *Disk(p, R)* defines the area of the disk with center *p* and radius *R*. The parameter *R* is specific to each application and accuracy requirements to localization.

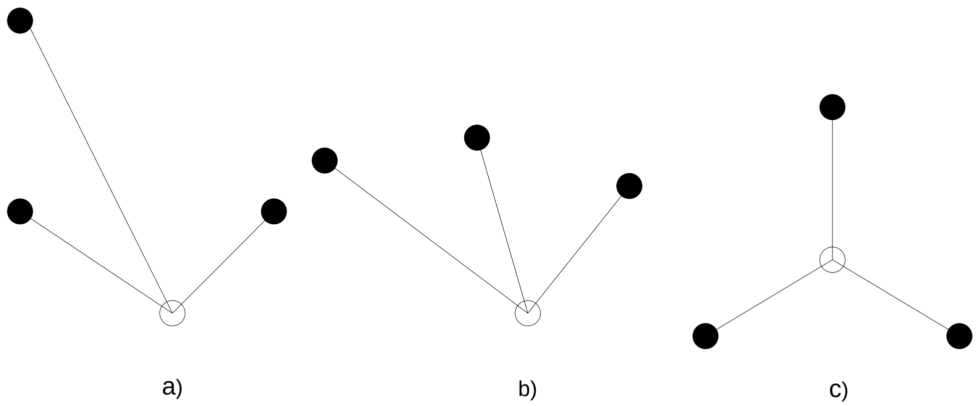


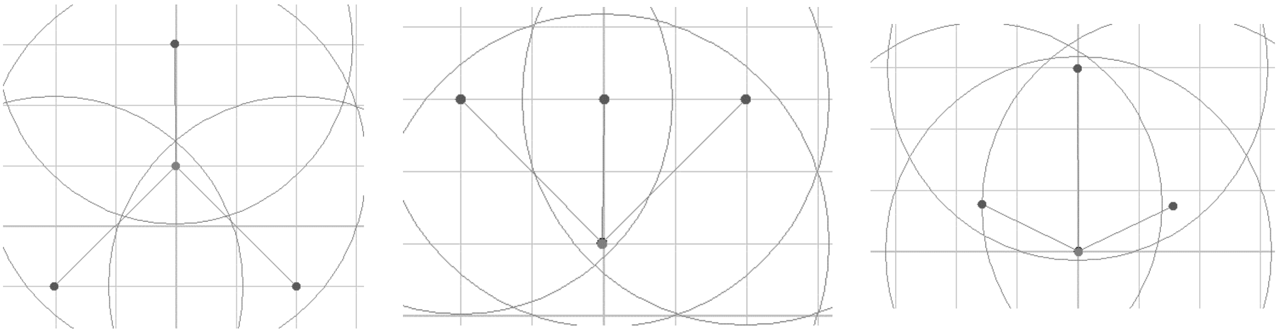

***Fig. 17.*** *Three sample scenarios of trilateration*

Fig. 17, (a), (b) and (c) shows three trilateration scenarios. Assuming a normal probability distribution of measurement error, the first scenario (a) shows the most common type of scene configuration. The second scenario (b) shows an extreme case, when all three reference points are approximately on one line and there is a high probability of such errors as a mirror reflection of the localized node. The third scenario (c) shows the ideal localization conditions when the anchor points are equidistant from a localized node and from each other.

### 3.2 Trilateration error modeling

To analyze and assess the impact of the reference point location scenarios against which the desired coordinates are calculated, as well as the parameters of the probability distribution and the nature of the error a series of experiments were planned and conducted. As a simulation environment, a standard software platform Java was chosen with additional charting library JFreeChart.

Three previously considered localization scenarios were implemented (Fig. 18), the ideal (a), pessimistic one (b), and averaged sample (c). For comparative evaluation three algorithm were selected supporting localization on three reference points: cluster algorithm, min-max algorithm and Levenberg-Markarth implementation of least squares method. As a result of the first iteration of the simulation the best of the algorithms with the greatest precision was selected, also taking into consideration execution speed. Then, for each of the scenarios localization error was modeled using Gaussian probability distribution, and finally the combined distribution of errors and their impact on the resulting localization error evaluated.



(a) (b) (c)

***Fig. 18.*** *Localization scenarios models*

Plan the first iteration of the simulation shown in Table 3. The types and number of error distribution settings correspond to the types of distributions. Since in this case when only the overall effect of the scenario on the properties of the resulting distribution of error was investigated, exact numerical values are not significant and numerical parameters of the distributions are selected based on the ease of demonstration.

Preliminary evaluation of the accuracy and speed of the algorithm was carried out taking into account the measurement error that has a Gaussian distribution with zero mean and a variance equal to 15. Totally 10,000 samples taken random variable has provided a sufficiently smooth probability distribution function (Fig. 19).

Nine starting experiments showed that while the execution time of considered algorithms, regardless of the anchor location scenarios differs insignificantly. Clustering algorithm was for about five to ten percent slower than the other two. For this reason, execution time of the algorithm further will not be considered.

***Table 3.*** *Simulation plan*

| № | Scenario | Trilateration algorithm | Error distribution | Parameters | Elapsed time |
|---|---|---|---|---|---|
| **1** | A | Cluster | Gauss | mean 0, var 15 | 31.59 (/4) |
| **2** | B | Cluster | Gauss | mean 0, var 15 | 31.38 (/4) |
| **3** | C | Cluster | Gauss | mean 0, var 15 | 30.12 (/4) |
| **4** | A | LS | Gauss | mean 0, var 15 | 30.25 (/4) |
| **5** | B | LS | Gauss | mean 0, var 15 | 26.41 (/4) |
| **6** | C | LS | Gauss | mean 0, var 15 | 27.14 (/4) |
| **7** | A | Min-Max | Gauss | mean 0, var 15 | 25.45 (/4) |
| **8** | B | Min-Max | Gauss | mean 0, var 15 | 26.48 (/4) |
| **9** | C | Min-Max | Gauss | mean 0, var 15 | 26.47 (/4) |
| **10** | A | Cluster | Gauss | mean 50, var 15 | - |
| **11** | C | Cluster | Gauss | mean 50, var 15 | - |
| **12** | A | Cluster | Gauss+Exp | mean 10, var 2, lambda 0.2 | - |
| **13** | B | Cluster | Gauss+Exp | mean 10, var 2, lambda 0.2 | - |
| **14** | C | Cluster | Gauss+Exp | mean 10, var 2, lambda 0.2 | - |
| **15** | A | Cluster | Exp | lambda 0.2 | - |
| **16** | C | Cluster | Exp | lambda 0.2 | - |
| **17** | A | Cluster | Rayleigh | var 8 | - |

Least squares algorithm and clustering algorithm showed approximately the same location accuracy (within statistical error) for an ideal (A) and conventional (C) scenarios. However, for a bad scenario (B), a cluster algorithm was more stable and about 50% of the cases provided successful localization results, while the least squares method stable fell into a mirrored location situation.

In case of large measurement errors (Min-Max algorithm) or incorrect localization (mirror location), the distribution of the error localization becomes Rice distribution, which corresponds to its definition: “If X and Y are independent Gaussian random variables with non-zero expectations generally unequal, Rayleigh distribution becomes Rice distribution”.

In the particular case, if we define equal non-zero expectations for Gaussian random variables whose localization error distribution will vary from Rayleigh distribution to Rice distribution depending on the anchors location scenario as it changes from the ideal to the poor. (Experiments 10, 11 - 50 Gauss, 15) Since the size of the localization result error indicates the direction (vector), for error filtering and verifying the results in each coordinate error individually should be used.

Min-Max algorithm showed ten times less accuracy than the first two, and for this reason in the future will not be considered. Further, only the cluster algorithm will be consider as more accurate and, at the same time, in contrast to the method of least squares stable with respect to bad scenario.

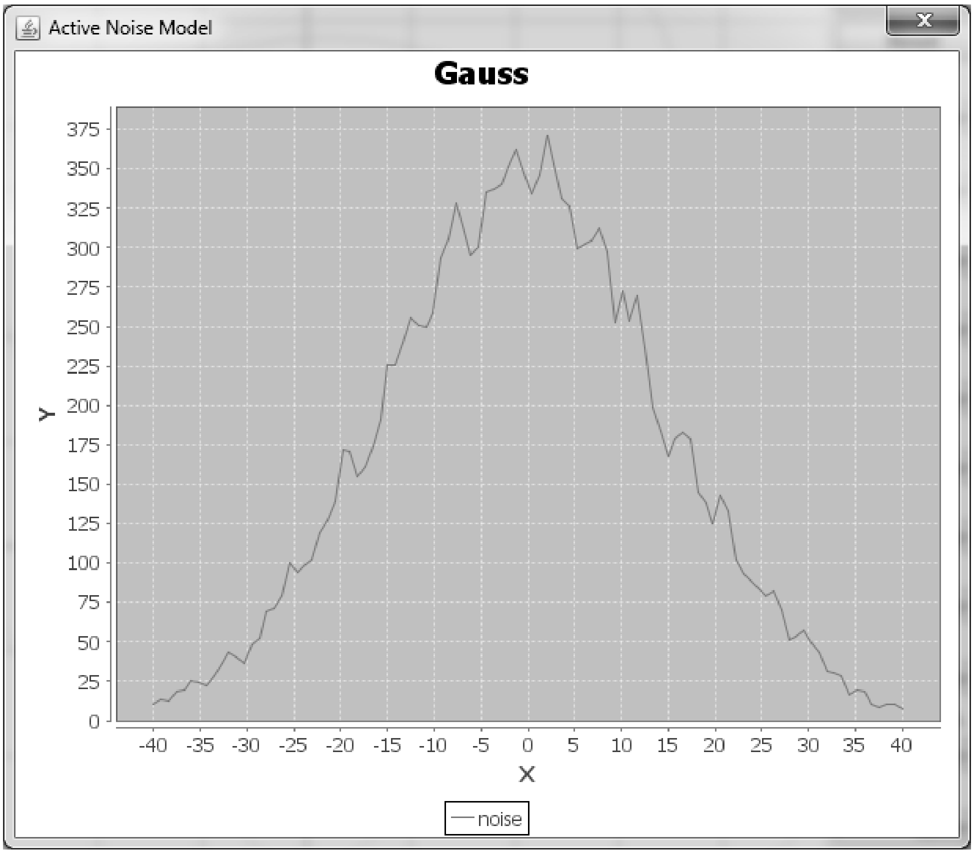



***Fig. 19.*** *Model with Gauss distribution*

### 3.3 Measurement errors comparison

Independent error estimates of each of the *x* and *y* coordinates have the same distribution law and the same parameters as the distance measurement error (Fig. 20).

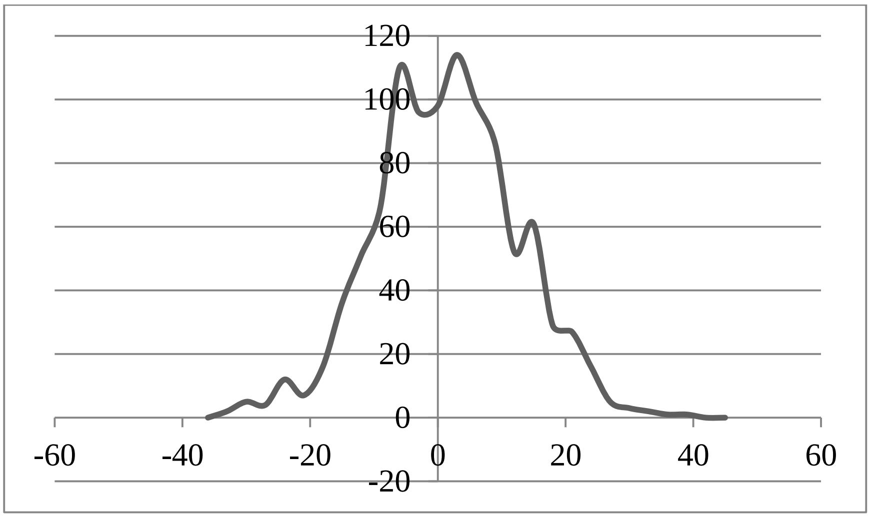



***Fig. 20.*** *Independent error distribution by x u y coordinates for scenario A*

Localization error (distance between the actual point and the resulting localization pint) correspond to the Rayleigh distribution (Fig. 21), which is consistent with the definition of "If X and Y are independent Gaussian random variables with zero mean and equal variances, the total distribution of the random variable is Rayleigh: $= \sqrt{x^2 + y^2}$ ”.

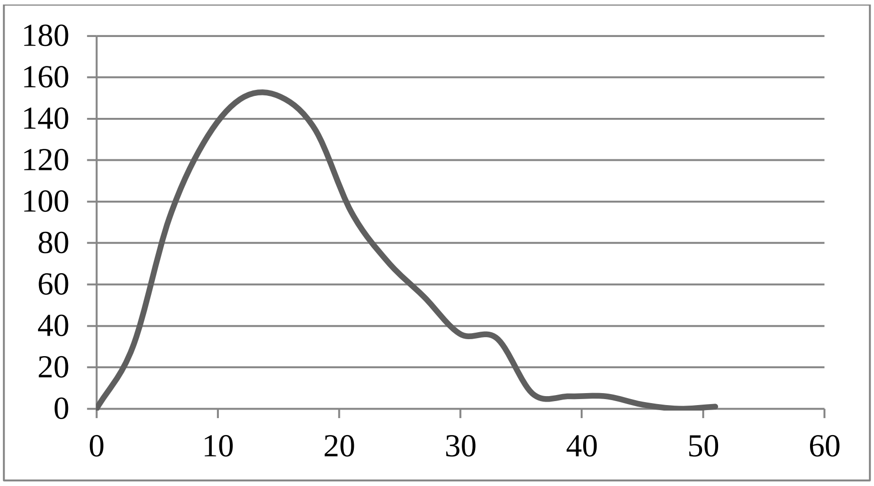



***Fig. 21.*** *Resulting error distribution for scenario A*

In cases of non-ideal scenario localization error probability distribution of each of the coordinates *x*, *y* (Fig. 22 and 23) can vary significantly, while, as the resultant probability distribution of error localization (Fig. 24) may remain constant.

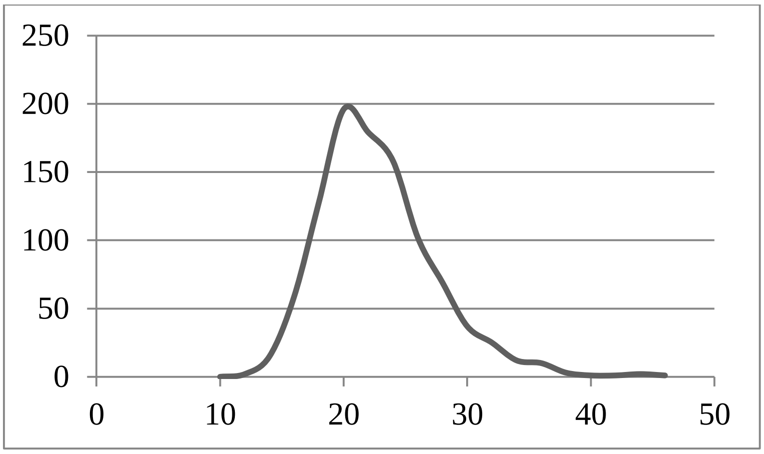



***Fig. 22.*** *Error distribution by x coordinate for scenario C*

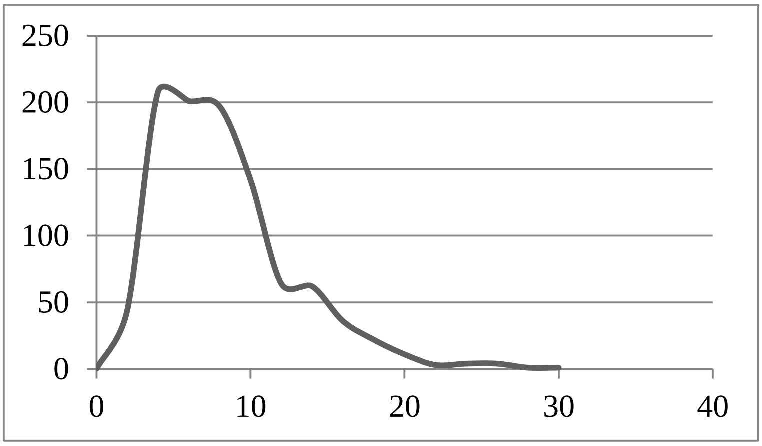



***Fig. 23.*** *Error distribution by y coordinate for scenario C*

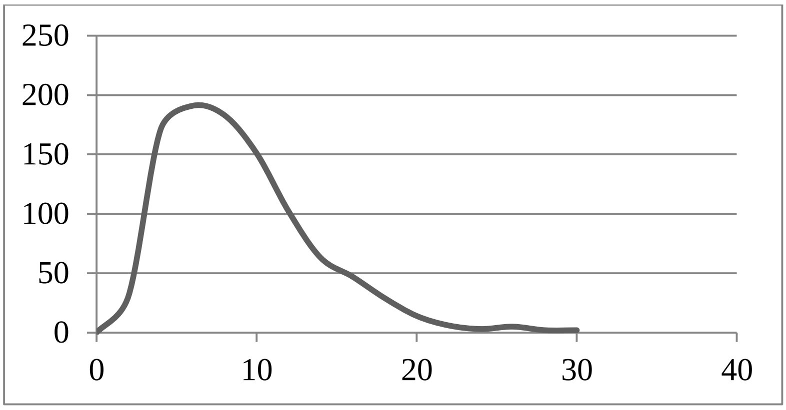



*__Fig. 24.__ Resulting error distribution for scenario C*

Since the distribution of localization error for each of the coordinates is “mixed” distribution of arithmetic sums of the errors distributions of measured distances, for better understanding the components of the overall localization error separate experiments were carried out to localize with distance measurement errors distributed both exponentially and with Gaussian distribution having the mean different from zero.

Accuracy estimation algorithms performed using the error consisting of the sum of a Gaussian distribution with mean equal to 10 and variance equal to 2 and the exponential distribution with parameter lambda equal to 0.2. Totally 20,000 samples of random variable has been taken (Fig. 25).

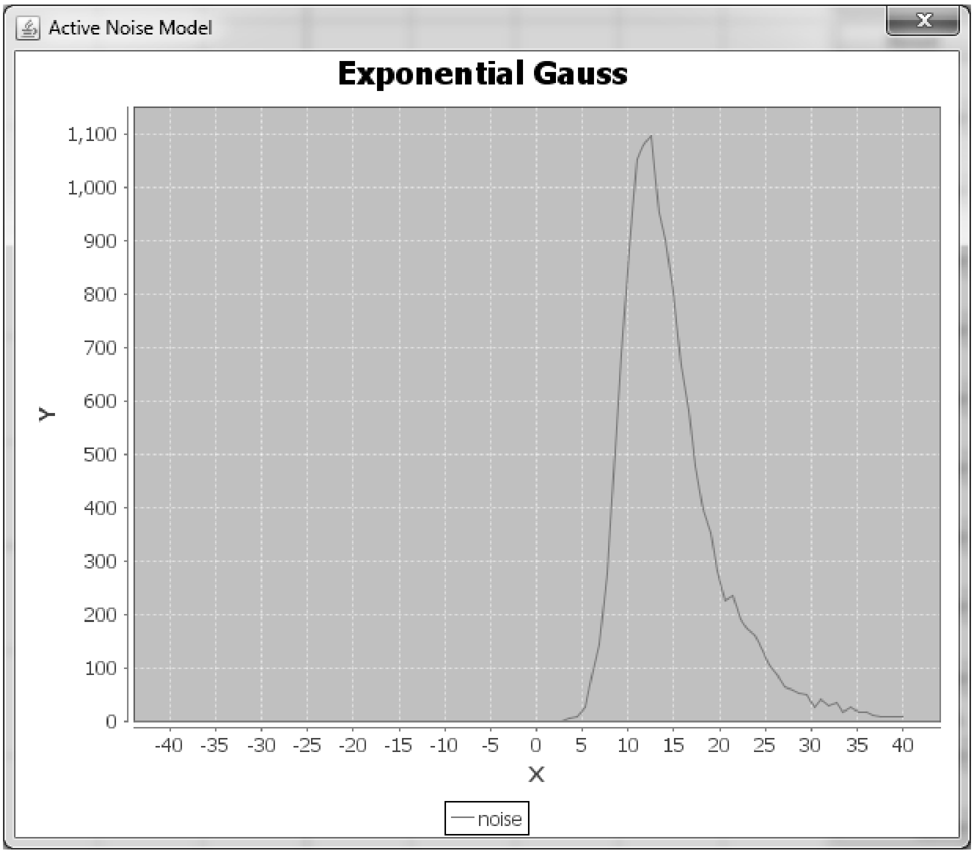



*__Fig. 25.__ Model with combined error distribution*

Experiments showed that in this case the total localization error always distributed according to Reiss, while the distribution of errors for each coordinate separately, depending on the scenario can be either Gaussian distribution (in the ideal scenario) or Rayleigh distribution (in bad scenarios). A similar pattern observed in case when distance measurement error distributed by Rayleigh.

### 3.4 Conclusions

Errors distribution in the case of localized coordinate with Gaussian distribution has the same parameters as the distribution of the distance measurement error. Other measurement error distributions, in localized coordinates, depending on the scenario, become either Gaussian distribution or a Rayleigh distribution and are summed. The total localization error (the distance between the calculated and the actual location of the object) has a lower value for the study because it does not indicate the direction (vector) localization of error, and has a Rayleigh distribution or Rice depending on the scenario in the case of the a non-zero expectation.

Simulation has shown that superposition of random errors in the localization process has the same nature as the superposition of harmonic oscillations with a random phase. It is well studied and has a developed mathematical description, what, if necessary, can be used in tasks of the distribution of localization errors analysis.

This section gives the characteristics of network nodes localization errors arising from inaccuracies in the measurement of distances to the reference points. The influence of the relative positions of reference points on the resulting error localization has been studied. Created model and a number of experiments carried out to investigate the properties of the probability distribution of the resulting localization error depending on the probability distributions of the error distance to reference points. Tests of behavior performed not only with a conventional normal distribution and the exponential distribution, but also with some combinations of different types of error distributions.

# 4. SCENE ANALYSIS ALGORITHM

This chapter presents a case study of localization of wireless sensor network based on anchor nodes and the matrix of distances between network nodes. Next, presents some basic concepts related to problems such as the graphs rigidity. Specially noted similarities of graphs embedding tasks with the task of the graph nodes localization. After a formal statement of the problem, proposes a method of placement and localization of the graph with additional sources of information. Given an example of a possible implementation, estimated overall performance and computational complexity of the proposed algorithm.

## 4.1 Euclidian distance propagation

Propagation localization method based on the Euclidean distance to a reference point propagation, which is similar to the method used in GPS localization. Selected node *A* must have at least two neighbors *B* and *C*, for which the estimation of their distances to the reference point *L* are known (Fig. 26). In addition, node *A* has a measured distance *AB*, *AC* and *BC*, what means that the conditions that *B* and *C* are neighbor nodes and can measure the distance between them are met, or the distance *BC* was known to node *A* a priori. This condition allows to position neighbor of A in the local coordinate system.

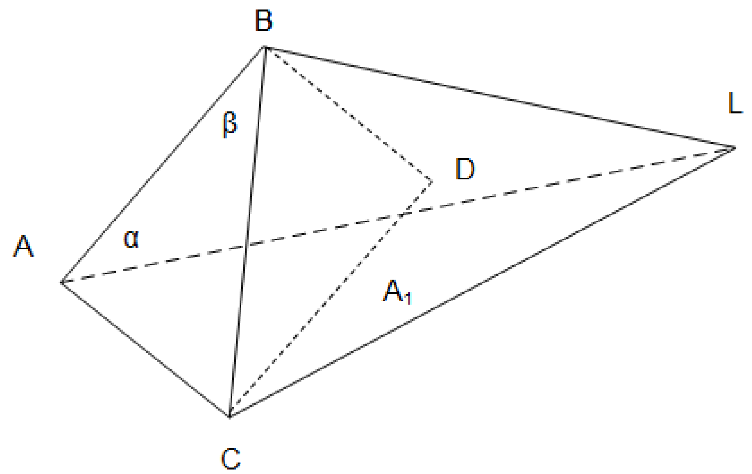



***Fig. 26.*** *Distance estimation from A to L*

In any case, we get a quadrilateral *ABCL*, all four sides of which and *BC* diagonal are known. This allows node *A* calculating the diagonal *AL*, which is actually a Euclidean distance from the node *A* to the reference point *L*.

There is a possibility that the node *A* is located on the same side of the *BC* line, and that the point *L* (in the figure indicated as $A_1$), and the distance to *L* has another value. The choice between the possible options is done either locally at the node *A*, based on the additional information received from other neighboring nodes, or by examining the relationship of nodes *B* and $C$ and their close neighbors. The latter case requires advanced communication capabilities between nodes. In this case, making a decision about the true position of $A_1$, node $A$ can use the distance to points $L$, calculated by other nodes that have no direct communication with $A$.

In the case of when it is impossible to make a clear choice between $A$ and $A_1$, distance estimation to $L$ remains inaccessible up to the moment when additional data from other nodes in the network is received. Once the right choice between $A$ and $A_1$ becomes possible, the distance is calculated based on the Pythagorean theorem for triangles $ACB$, $BCL$, and the $ACL$, what allows to determine the length of the $AL$. Distribution of Euclidean distance allows the use of methods reducing errors except for those that may be contained in the $GCP$.

$$\cos\alpha = \frac{AB^2 - AC^2 - BB^2}{2\cdot AC\cdot BC}$$
$$\cos\beta = \frac{BL^2 - BC^2 - CL^2}{2\cdot CL\cdot BC} \tag{25}$$
$$AL^2 = AC^2 + CL^2 - 2\cdot AC\cdot CL\cdot\cos\alpha\pm\beta$$

$$\sigma_{AL}^2 = \sum\left(\frac{\partial AL}{\partial e}\right)\sigma_e^2{}^2 \tag{26}$$
$$e = AC, CL, LB, BA, BC.$$

A useful feature of the algorithm also is the fact that if we know the proportion of uncertainty in all the measured distances, the percentage uncertainty in the $AL$ can be calculated in the process of propagation. This property allows reducing the error in the location estimate, up to 50%. Part of uncertainty $\sigma AL$ propagates further to other nodes that use the calculated distance to the $L$ in its own localization [131].

### 4.2 Graph embedding

Another approach to the problem of localization is to consider it as similar to the problem of graphs embedding [132]. Obviously, for the nodes in the proximity of the reference point, the distance to this point can be estimated by direct measurement. Using certain methods of coordinate propagation distance to the reference point can be estimated for the second level of the neighbors, separated by two hops. Thus, by traversing entire network can be examined. If the graph is sufficiently connected and the lengths of its edges are known, it could be reconstructed on the plane.

Thus, the problem of localization can be viewed as the problem of reconstructing a set of locations of sensors based on a variety of known distances between sensors [133], being within a certain radius around each sensor. In this case, some of the sensors can be anchors, but for the placement in local coordinate system it is not so important. At its core, the problem can be formulated as the problem of determining whether a given graph can be physically implemented in two-dimensional space.

In contrast to the classical methods of localization, suggesting that the network must contain a significant number of nodes anchors, embedding a graph with the coordinate propagation does not need anchors for the local coordinate systems and therefore requires much less anchors for binding to the global system [134]. All nodes can be initialized with random coordinates, which as a result of the algorithm execution and data exchange between nodes must match with coordinates that do not contradict the known distances.

The resultant coordinates will have certain degree of freedom in terms of orientation and translation, but the scale will be respected. Subsequent processing may include absolute positioning with reference to the three or four reference points, which makes it possible to eliminate the additional degrees of freedom. As possible set of constraints that reduces the number of required reference points, additional geolocation information can be used.

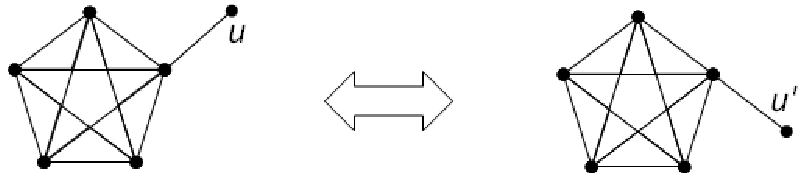
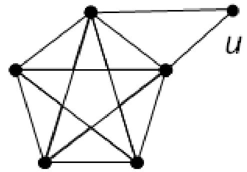
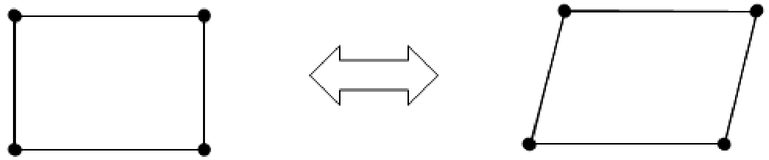
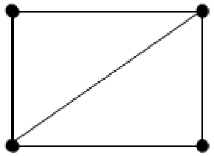
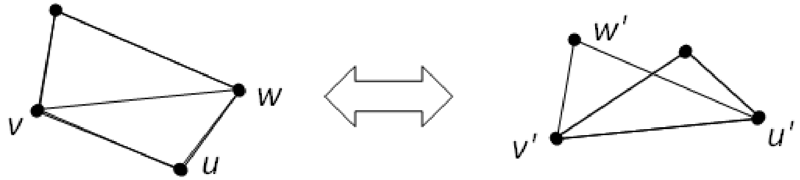
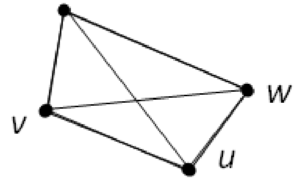
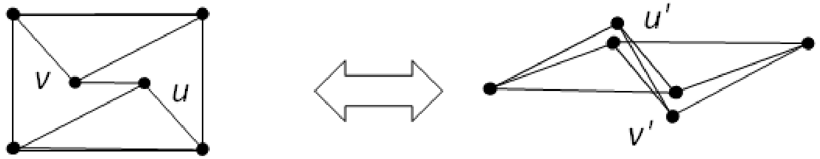
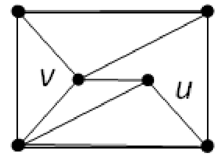
Formally, the problem of placement of the graph can be described as follows: placement of *G* on the surface $\Sigma$ is a projection of *G* on $\Sigma$, where point on the surface $\Sigma$ associated with the vertices of the graph *G*, but simple arcs of surface $\Sigma$ associated with edges of the graph *G*, moreover:

- The end point of the arc associated with edge *e*, are the points associated with the vertices of *e*;
- Neither arc does not include the points associated with other nodes;
- Two arc never intersect at a point that is not a final for both arches;

In other words, placing a graph is such its location on the surface, in which the edges of the graph intersect only at their endpoints.

Another problem is closely connected with the problem of graphs embedding and networks localization. This is the problem of determining the rigidity of the graph [130]. In the context of the graph *G* realization task it is considered as a function mapping vertices of *G* at points in two-dimensional Euclidean space. In general, the implementation of the graph is considered to be admissible if constraints are met in form of distances between pairs of vertices *i* and *j*, where the edge $(i, j) \in E$. This, as mentioned above, means that the $d(i, j) = ||(p\ (i) - p\ (j)||$ for all $(i, j) \in E$. Two realization of *G* are equivalent if they are identical for all symmetry operations: reflections, translations and rotation. Distances graph *G* should have at least one

***Table 4.*** *Graph deformation*

| Deformation | Non-Unique Graph realization | Solution | Resulting Graph |
|---|---|---|---|
| Continuous |  | G must be rigid |  |
| |  | |  |
| Flip |  | G must be 3 connected |  |
| Flex |  | G must be redundantly rigid |  |

consistent realization that corresponds to the actual location of the topological network. This implies that *G* is connected and has at least four vertices.

Graph is generically rigid if it cannot be deformed continuously in any of its realization, without violating distances constraints [135, 136]. Graph is generically globally rigid if in translation, reflection, rotation has only one unique realization (Table 4). Realization of graph considered common in the case where the coordinates of its vertices are algebraically independent. Here and in further on we will only talk about the common realization of graphs.

There are several types of way how not unique realizations of graph may arise. Graph, which can be continuously deformed satisfying all restrictions, is flexible, otherwise rigid. Thus, the rigidity of the graph is a prerequisite for its global rigidity. Nevertheless, in some circumstances, rigid graphs can also be subjected to deformation. For example, an uncertainty of reflection when a subset of vertices has two possible mirror realizations. This type of uncertainty is not available for graphs in which every vertex is connected to at least three other vertices.

Suppose that after the removal of one edge from three connected vertices, other configurations becomes possible for sub graph satisfying all constraints after recovery of deleted edge. In case when it is impossible for every graph edge, we can speak of a redundant rigidity, meaning that if you remove any of the edges, the graph remains rigid.

Summarizing conditions precluding uncertainty in the placement of graphs, Jackson and Jordan proposed the theory of necessary and sufficient conditions for the global rigidity of graphs.

**Theorem 1**. [137]. Graph with $n \geq 4$ vertices is globally rigid in two dimensions if and only if it is three connected and redundantly rigid.

Based on Theorem 1, the property of the global rigidity of the graph can be tested by polynomial time using Pebble algorithm [138] and network traversal algorithms [139].

### 4.3 Localization as graph embedding, problem formulation

Assume that *N* nodes labeled *1, ... N*, are located in unique locations in a physical region. Also, assume that there is some mechanism, which allows each node to discover neighboring nodes by using communications with these nodes. This mechanism in addition allows estimating the distance to the neighbor nodes. Each discovered neighboring node gives one undirected edge $e = (i, j)$ for a graph *G*, that describes known nodes.

Thus, given the graph consisting of $n$ vertices $G(V = \{1, ..., n\}, E)$, and its Euclidean length $l_{i,j}$ for each edge $(i, j) \in E$. Denote the two-placement arrangement of the graph as $x, y \in \mathbb{R}^n$ ^ n, where the coordinates of vertex $i$ are denoted as $p_i = (x_i, y_i)$, and the distance between the vertices $i$ and $j$ as

$$d_{ij} = \|p_i - p_j\| = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{27}$$

Not yet taking into account the measurement error, we can say that there is a physical placement of network nodes, which implements given length of edges of the distances graph $d_{i,j} = l_{i,j}$. The aim is to reproduce this placement, and, placement corresponding to these requirements may not be unique.

Not unique placement is acceptable within such operations as rotation, translation and reflection, however, the measured distance must clearly fix placement scale.

For some graphs configurations placement can be non-unique, and have the possibility of transformation, other than rotation, translation and reflection. In order to embed such graph adequately, it should have the rigidity that prevents the flex while preserving all the distances between vertices, but even rigid graph can have local flex areas. Thus, successful localization needs globally rigid graph with a unique placement options. The above is confirmed by the theorem given in [140].

**Theorem 2.** Assume that $N$ network located in the coordinate system $\mathbb{R}^d$, where the number of dimensions $d = 2$ or $d = 3$, is composed of $m > 0$ anchors located at points $p_{m+1}, p_{m+2}, ... p_n$, and $n - m > 0$ nodes with unknown coordinates at the points $p_{m+1}, p_{m+2}, ... p_n$. For the case where $d = 2$, suppose that there are at least three anchor nodes. Accordingly, in case where $d = 3$ at least four anchors. We denote as $\mathbb{F}_p$ points formation corresponding to the position of nodes $p_1, p_2, ... p_n$, and links between the points marked as pairs of neighbor nodes and anchors in $N$. For $d = 2$ and $d = 3$ cases, the problem of localization can be solved if and only if the formation $\mathbb{F}_p$ globally rigid. Thus, all nodes in the globally rigid subgraph having at least three anchors are localizable.

Obviously, in the case where the number of anchors is insufficient, or when the formation is not globally rigid, it is necessary to seek and to use other possible sources of information to avoid improper placement of the graph and to find localization task solutions.

In addition, please note that in the real world the measured distance $l_{i,j}$ between network nodes contain an error component $l_{i,j} = d_{i,j} + \varepsilon_{i,j}$. This means that the optimal placement solution might not exist. In such cases, there is an additional problem to minimize the difference between the actual deployment of network and topological placement, obtained by calculations.

## 4.4 Localization with additional information resources

Digital maps that have already become widely available location-based resource that carry a lot of potential additional source of data for the object localization methods in wireless transportation networks. Using raster data type in such problems is not a trivial task, but using vector data type in the task of distance graphs placement on the ground can be successful and without significant additional costs. For example, a vector data format shapefile [141], is widely used in today's geo-location information systems, based on the data structures that describes the geometry of the type of points, lines, arcs, polygons, etc., and on separate layers stores additional information attributes and bitmaps .

These sort of data viewed in the context of road infrastructure, can provide substantial assistance to improve localization processes in transport, excluding those locations at which the vehicle cannot be located physically as well as to refine the results of localization, comparing them to known geometric arrangement of the roadway. In a sense, the vector component of the digital maps can be considered as the global graph, inside which the vertices of the subgraph formed by the wireless network nodes are placed, while vertices of the subgraph can or cannot match with vertices of the global graph, and can be placed on its edges.

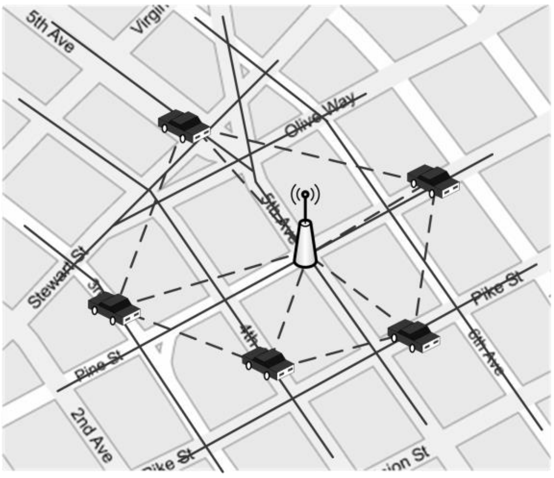



***Fig. 27.*** *Vertices of distance graph embedded into geo-location data graph*

Fig. 27 shows a graph of distances formed by distances measured between neighbor vehicles, as well as the distances to the preset infrastructure anchors with known coordinates. This graph should be embedded in the graph formed by the street digital map so that all vehicles are in line with its edges or vertices. Using these additional restrictions, in some cases, objects can be located, even if the graph of distances is not rigid by itself. Moreover, even acyclic graphs (trees), with such a method may be placed correctly, unlike with other previously discussed classical techniques.

Another significant advantage of this approach is the possibility of successful placement with less and formally insufficient number of anchors. Essentially, to determine the search window for embedding into global graph one anchor can be sufficient.

At the same time when the majority of localization algorithms for sensor networks do not consider the mobility of nodes in an explicit form, meaning that the network is static, wireless networks in transport, as a rule, have a dynamically changing topology. However, this disadvantage can be turned into advantage in the case when vehicles uses dead reckoning system [142], taking into account the information about the direction, speed, acceleration, etc. Preserving in memory localization details, the last measured distance to the node anchors, as well as taking into account new dimensions and configuration of the distance traveled, the graph distances can be extended with new nodes and edges. Additional nodes would improve rigidity and connectivity of the graph, thus adding new constraints and significantly increasing the probability of successful localization and absence of alternative placements in the global digital map graph.

Using the previously recorded position of the vehicle as additional constraints does not require the localization algorithm some special calculations other than the calculations over the current position. Anchor node or the node having direct communication with the anchor, can calculate the location of their immediate neighbors in iterative manner regardless of whether they are new nodes that have emerged in the line of sight, or are recorded historical data about the location of network nodes.

Another assumption about the geometry of the road, that can be made to improve the localization algorithm, is as follows. As noted above, the vector map of the location of all the objects usually expressed as a set of pairs of values of *X*, *Y* or, in the case of three-dimensional map, values of *X*, *Y*, *Z*, a coordinate system used in the map. Location-based properties of the objects described in the form of points, lines, arcs, and polylines. Each particular implementation of the data structure, however, largely depends on the manufacturer of the digital map and what standards are supported therein. In the discussed case, all that is required for the localization algorithm as input, is a list of edges of the graph and the software procedure, the result of which indicates whether an investigated point matches with the selected edge from the list or not. The degree of connectedness of the graph and the shape of the edges in this case does not affect the logic of the algorithm. This fact allows to simplify the localization algorithm simulation and debugging, reducing the geometry of the road up to the simplest grid. In such a grid, each edge from the list and the coordinates of the point can serve as arguments for the function testing of their match with a given tolerance. Modeling should take into account measurement errors, a non-zero value of the components of road infrastructure and the size of the vehicle.

## 4.5 Embedding algorithm

The main idea of the approach based on localization with iterative placement of the graph is to use the known data on the coordinates of nearby landmarks together with the coordinates of known anchor nodes to localize the remaining may not sufficiently connected with each other anchor and nodes with unknown coordinates. Without any loss of applicability, this approach fits static network located in two-dimensional space. For simplicity, first let us consider the case where each sensor of the network has the ability to measure the distance to all other network sensors, regardless of the distance.

Given distances graph *G(V, E)* and the graph of digital maps *I(W, F)*. To start the algorithm, at least one node *n* in the network 1, ..., *N* must be an anchor $b_0 \in N$ with known coordinates $x_0, y_0$ in the chosen coordinate system.

We use the definition of "neighbor" to denote the set of nodes $v \in N$, whose distance from the starting anchors node is known and available for computational processes. Another definition that is required to introduce is a reflection nodes *s*, meaning the whole set of points of intersection of graph elements of digital map $\{s | s \in \mathrm{W} \vee s \in \mathrm{F}\}$ and the circle circumscribed around the anchor node $b_0$ with coordinates $x_0, y_0$. The diameter of the circle $R$ assumed to be equal to the distance *d* node *v* being localized.

Coordinates of *s* given by:

$$\begin{cases} x = x_o + R\cos\varphi \\ y = y_o + R\sin\varphi \end{cases} \quad 0 \le \varphi < 2\pi \tag{28}$$

The physical meaning of this definition is to display all valid physical location of this node in the existing topological context (in this first example, the simple grid).

Given the above, the algorithm is as follows:

1. For each anchor node *b*, select all neighboring nodes *v* and group them based on whether there exists measured distances between each *v* in the group (connected nodes). In the case of unconnected nodes group can consist of a single node, and when all nodes are connected, respectively, form only one group (Figure 29);
2. In each group for each node, calculate the coordinates of its reflections $s_n$, by finding the points of intersection of the graph elements *I*, and a circle with the center with coordinates of the node $v_n$ and a diameter equal to the known distance from that node to anchor $b_n$. Step 2 of the algorithm illustrated by Fig. 30;
3. For each node in the group $v_n$ calculate the coordinates of the set of possible reflections $s_n$;
4. If the group contains more than one node for each pair $s_n$ $s_{n+1}$ check whether the distance between the known distance between $v_n$ and $v_{n+1}$;

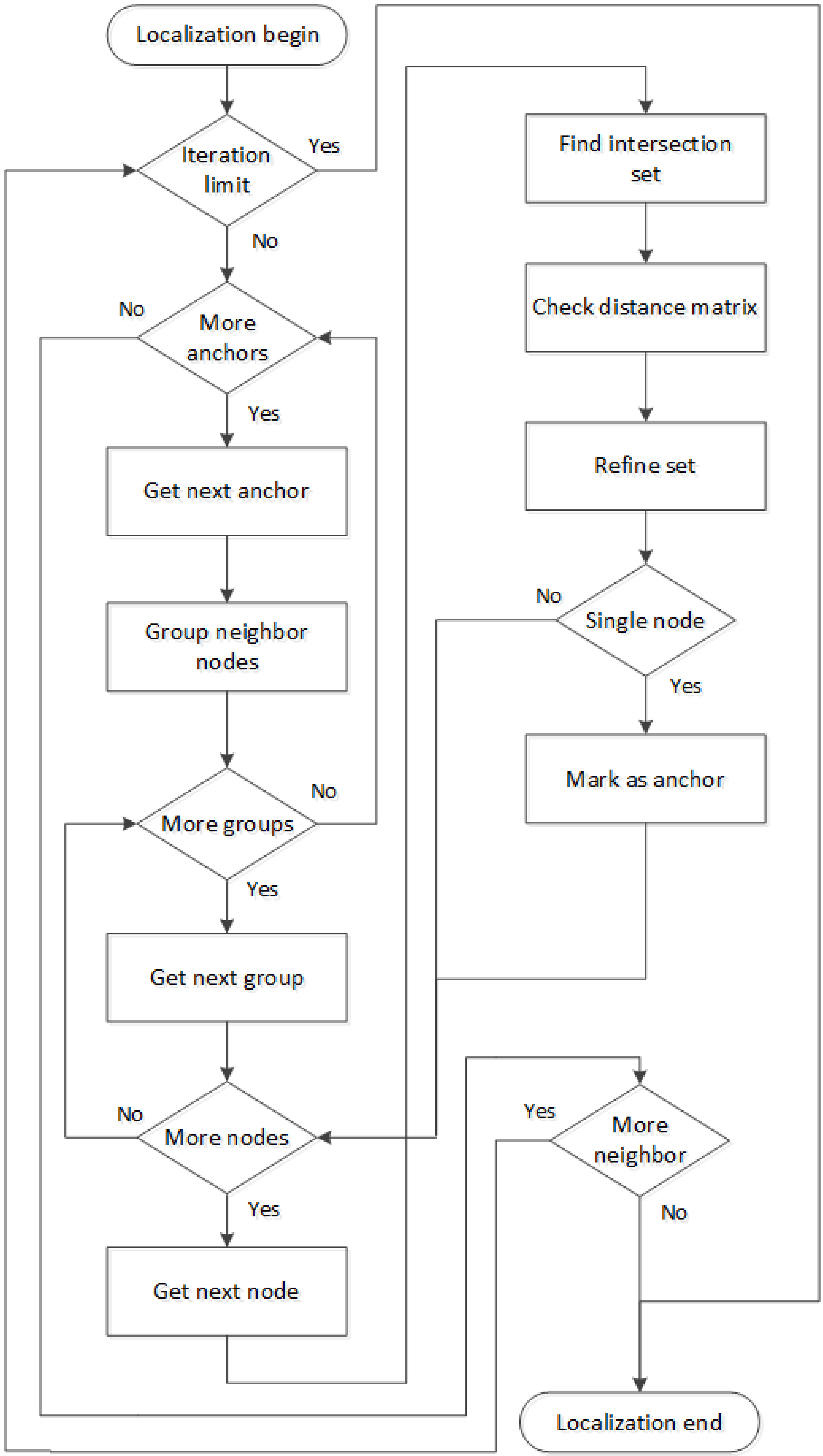



***Fig. 28.*** *Localization algorithm scheme*

5. Exclude from node combinations nodes that do not match a known distance measurement, taking into account a given by a model certain tolerance to errors;
6. If the group remains the only one combination of reflections $s_n$, $s_{n+1}$,..,$s_{n+m}$, the nodes in the group are localized (Fig. 31);

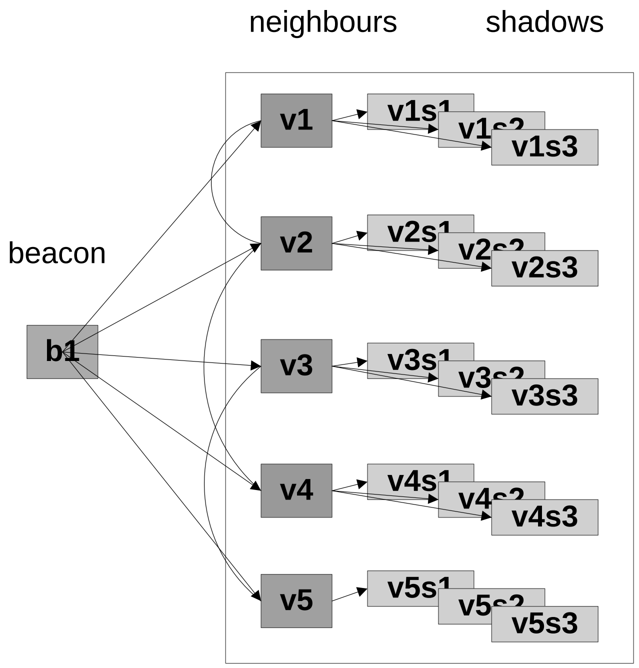



***Fig. 29.*** *Algorithm initial data structure*

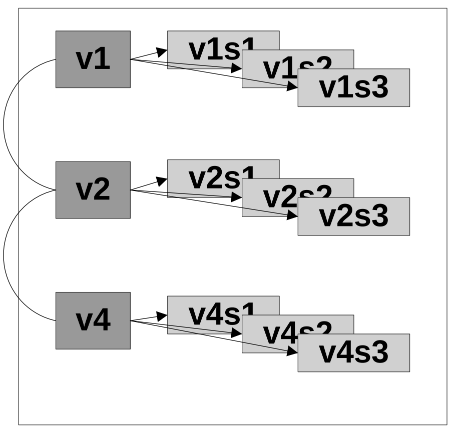



***Fig. 30.*** *Grouped nodes*

7. The algorithm repeated for anchors and nodes, localized at the previous iterations of the algorithm and, from this moment, also considered as additional anchors, until any anchor has at least one neighboring node that is not localized. Stopping criteria of the algorithm is the localization of all nodes in the network or the reaching a predetermined maximum number of repetitions.

General block diagram shown in Fig. 28.

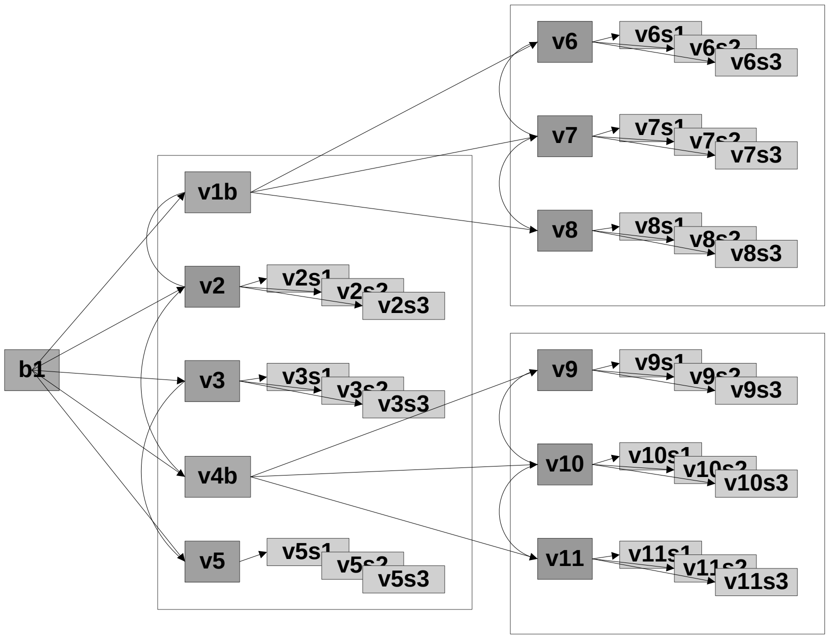



***Fig. 31.*** *Second iteration data structure*

### 4.5.1 Modeling environment settings

Table 5 demonstrates pseudo-code, which gives an overview of the localization algorithm. Functions `find_neighbors` (find neighboring nodes), `split_to_groups` (split nodes into groups), `compute_all_combinations` (find all nodes combinations) and `evaluate_shadow_layout` (evaluate placement reflections) described above are trivial operations of computational geometry and combinatory.

Since the considered modeling stage has been tasked with the lowest cost just to check and to visualize the algorithm to better understand the logic of his work, HTML5 language has been chosen as a platform. The choice was motivated by the fact that in HTML5 realized numerous new syntactic features such as elements <canvas>, the possibility of using SVG (Scalable Vector Graphics) [143] and mathematical formulas. These innovations are designed to simplify the creation and management of graphical objects, without the need for third-party API.

For algorithm simulation and debugging the following settings were selected: a network of eleven nodes, ten of which are not localized, and each has a measured distance for up to six neighbor nodes, one anchor node and grid, representing in this case the elements of a digital map.

The Fig. 32 shows the third step of the algorithm, where node number 11 is the anchor, and the nodes 1, 2, 3, 4, 6, 8 and 9 have been calculated and are marked on their reflection grating.

After performing the first iteration of the algorithm unique placement has been found for six not-located nodes that in turn became anchor nodes. The second iteration led to the localization of three more nodes (4, 7 and 10), and the third iteration of step 3 which is shown in Fig. 33, localized the last non-localized node 10.

***Table 5.*** *Localization algorithm pseudo code*

```
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);
         }
      }
   }
  }
}
```

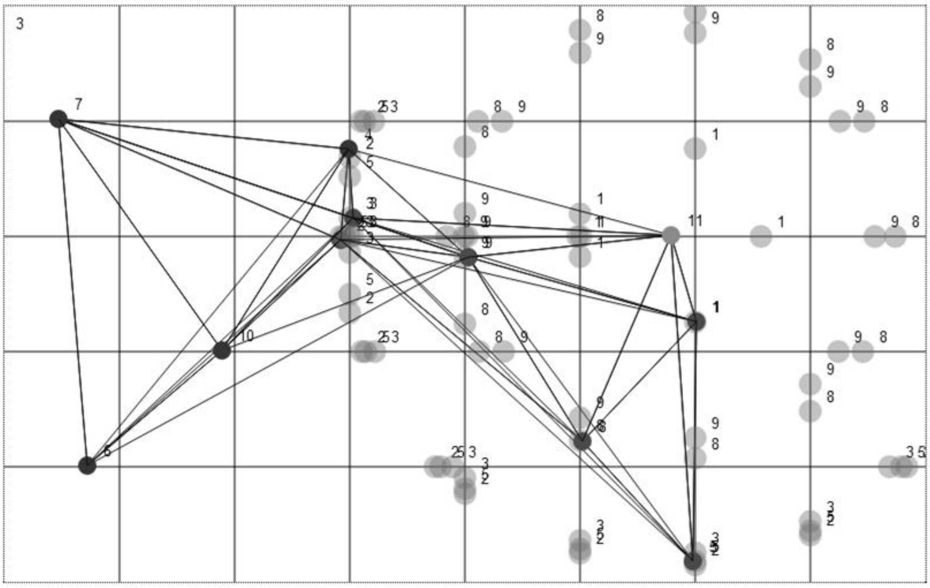


***Fig. 32.*** *Algorithm simulation, step 3*

In the case of a static environment and poor placement scenario, some components may remain non-localized, and the algorithm must stop after pre-determined stop condition will be achieved. Nevertheless, for the dynamic scenario, it makes sense to continue the algorithm. Nodes locations changes can lead to such placement, which will not have alternatives, or to give an opportunity to get new range measurement between nodes. In the case where nodes are equipped with dead reckoning, distance graph can be expanded memorizing historical node locations, and thus getting the new restrictions, which in turn will make it possible to find a unique placement. In addition, the dynamic scenario new members of the network may arrive, what significantly increases the probability of successful localization.

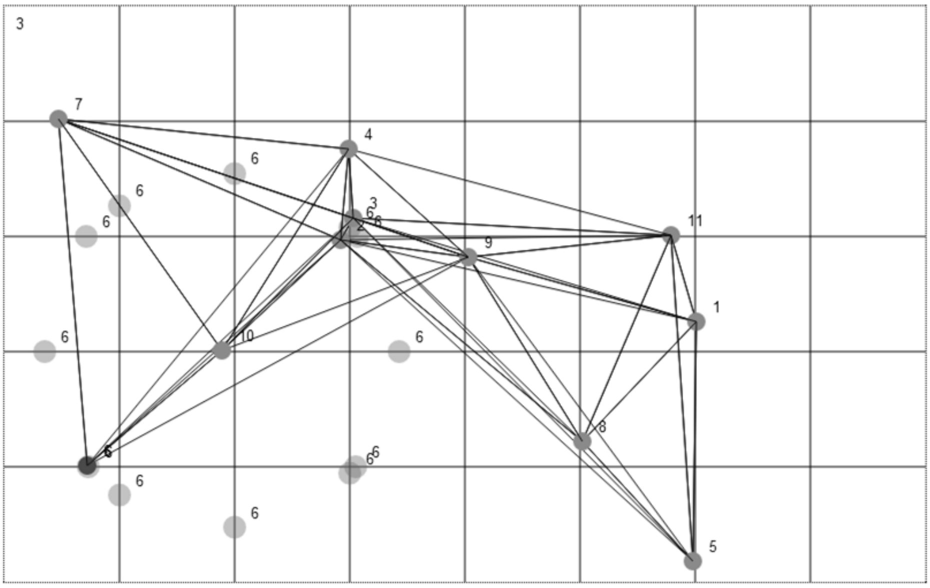



***Fig. 33.*** *Algorithm simulation, the last non-localized node*

Another thing that requires mentioning is the fact that in this case the goal was only in the evaluation of performance of the algorithm itself, and considers only the simplified localization scenario using the grid as a digital map. Such a scenario is actually quite pessimistic, since this configuration has a high degree of symmetry and is prone to alternative graph placements of type rotation, translation and reflection. Further a more realistic scenario will be considered, using actual examples taken from digital maps are typically characterized by greater asymmetry.

### 4.5.2 Computational complexity

In general, the complexity of the algorithm can be estimated by a function depending on the volume of work performed by algorithm and the size of the input data. The algorithm has a complexity of *O(f (n))*, if the when input size incises to $N$ dimension , the running time is increased at the same rate as the function *f(N)*. Consider the code provided in Table 6 that finds maximal element in each row in a matrix $A[N\times N]$.

In this algorithm, the variable $i$ vary from $1$ to $N$. At each change of $i$, $j$ variable also varies from $1$ to $N$. In each of $N$ iterations of the outer loop, the inner loop is executed $N$ times, too. The total number of inner loop iterations equals $N * N$. This determines the complexity of the algorithm $O(N^2)$.

Estimating the order of complexity of the algorithm, it is necessary to use only the part that grows the fastest. Assume that the working cycle described by expression $N^3+N$. In this case, the complexity is equal to $O(N^3)$. Consideration of the fastest-growing part of the function allows evaluating the behavior of the algorithm with increasing $N$. Calculating O constant factors in the expressions can be ignored. Algorithm with a working step $3N^3$ treated as $O(N^3)$. This makes the dependence of the ratio $O(N)$ from the task size changes more obvious.

***Table 6.*** *Sample algorithm for computational complexity estimation*

```
for i:=1 to N do
begin
 max:=A[i,1];
 for j:=1 to N do
  begin
   if A[i,j]>max then
   max:=A[i,j]
  end;
 writeln(max);
end;
```

In the previous example, the entire algorithm is executed by two cycles. If one procedure calls another one, it is necessary more thoroughly estimate the overall complexity. If it executes a certain number of instructions (eg, printing), is virtually has no effect on complexity estimation. However, if the called procedure executes $O(N)$ steps, this function can significantly increase algorithm complexity. If the procedure is called inside the loop, the impact can be much greater.

***Table 7.*** *Simplified localization pseudo code*

```
for 1 to I // stop conditions are not met
begin
 for 1 to B // beacon_array length
  begin
  for 1 to G // neighbor_groups
   begin
   end;
  for 1 to N // possible_combinations
  begin
 end;
end;
end;
```

If the inner loop of one procedure there is a call to another procedure, the complexities of both procedures are multiplied. In this case, the complexity of the algorithm is: *O(N2) * O(N3) = O(N5)*. If the main program calls a procedure at a time, theirs complexities are added: *O(N2) + O(N3) = O(N3)*.

Based on the above, we estimate the computational complexity of the localization algorithm, which was built by the examined model first. For this purpose, we simplify the pseudo code given in Table 5, to cycles taking as example code given in Table 6. The simplified pseudo code is given in Table 7.

Since, in contrast to the example considered, the algorithm is not a function with one variable, the computational complexity is the sum of the number of repetitions $I$, the number of anchors $B$, the number of localized nodes $N$, and also depends on the number obtained in the process of computing node groups $G$ and reflections for each node $S$.

Strictly speaking, referring to the combination of the location of nodes in terms of combinatory we talk about combinations without repetition, the number of $n$ for which over $k$ is the binomial coefficient:

$$C_n^k = \frac{n!}{k!\,(n-k)!} \tag{29}$$

Consequently, the total number of combinations will be the sum of combinations of the reflections over maximum number of reflections for one node in all groups of connected nodes. In addition, given the fact that the presence of three or more anchors, localization becomes possible by classical methods, the number of anchors $B$ shall not be greater than two, and justified number of algorithm repetitions cannot be greater than the number of non-localized nodes. Consequently, the function of maximum computational complexity is as follows:

$$f(N) = N \cdot 2 \cdot \left(\frac{N}{G} + \frac{n!}{k!\,(n-k)!}\right) \tag{30}$$

Thus, the computational complexity depending on the connectivity of the graph (number of groups) can vary from $O(N^2)$ to $O(N!)$.

Table 8 lists the approximate execution time of different computational complexity samples for the computer running one million operations per second. It should be noted, an optimistic scenario of the proposed algorithm computational complexity $O(N^2)$ is acceptable, and pessimistic, the maximum variant computational complexity of $O(N!)$ is too expensive, and requires optimization algorithm. A possible way of this optimization will be discussed in the next chapter.

***Table 8.*** *Approximate calculation time (one million operations per second)*

| | N=10 | N=20 | N=30 | N=40 | N=50 |
|---|---|---|---|---|---|
| $N^2$ | 0.001s | 0.008s | 0.027s | 0.064s | 0.125s |
| $2^N$ | 0.001s | 1.05s | 17.9min | 1.29 day | 35.7 year |
| $3^N$ | 0.059s | 58.1min | 6.53 year | $3.86 \cdot 10^5$ year | $2.28 \cdot 10^{10}$ year |
| N! | 3.68s | $7.71 \cdot 10^5$year | $8.41 \cdot 10^{18}$ year | $2.59 \cdot 10^{34}$ year | $9.64 \cdot 10^{50}$ year |

### 4.6 Simulation results

It is obvious that the problem of localization, solved the above algorithm as a classical problem of placement of graphs is NP-complete in the class NP, to which can be reduced any of the problems solved in polynomial time [144]. As can be seen, the algorithm on its steps 4 and 5 actually makes an exhaustive search to avoid conflicting combinations of nodes. Unfortunately, over certain network size time of the exhaustive search can become prohibitively large.

On the other hand, along with a continuous increase in productivity of modern computers, large enough NP - complete problems have already become quite rapidly solved. For example, the traveling salesman problem with the complexity of 2000 cities, now is solved by brute force is an acceptable period of time [145]. And when the data is well structured, the complexity of the problem being solved can be increased up to 14000 cities. In other words, there is a certain distance between the modern empirical results of various implementations of algorithms and theoretical studies known to us.

Moreover, the algorithm presented can be performed in a distributed mode, on many computing devices in parallel mode and independently. Thus, it seems unlikely that in a real transportation system the number of processing nodes will reach a value that can take too long time handle combinations of nodes in groups and cause unacceptable delays.

Another serious challenge is to ensure adequate processing of the measurement errors and tolerance to non-zero dimensions of road infrastructure and vehicles. Given these additional facts, the result of decisions concerning the match or mismatch of node and of a digital map element may largely depend on the angle between this element and the vector distance. In case where angle is acute, the result will largely depend on the selected tolerance, and in some cases will show two or more intersection points, adding excessive possible node locations. Handling such situations may complicate the algorithm and generate conflicting results.

A possible solution to this problem is to modify the procedure only discussed above, the result of which indicates if point belongs to the digital map elements. This approach will split the search problem geometrical intersection and graph embedding thus leaving placement algorithm unchanged and with the same computational complexity as it was described earlier. At the same time, we should pay attention to the fact that meanwhile there was considered only a simplified model with the digital map in the form of grid. Further, more realistic scenarios will be considered, with fragments of maps, consisting of more complex geometries. Therefore, the functional tests should include and adequately handle such scenarios as well.

In this chapter, the general and theoretical problems in the task of wireless sensor networks localization were brought into action. It was pointed out the possible direction in solving such problems in cases where the classical methods of localization are not applicable. The conceptual idea of the localization method with additional information resources was proposed. Has been created a model to carry out algorithm preliminary verification, evaluate its performance and computational complexity, as well as to find possible drawbacks that require further development.

# 5. SPATIAL QUERIES OVER SENSOR NETWORKS

This chapter explores the possibility of using spatial queries to the data objects represented by geometric abstractions, as a tool applicable in localization problems for wireless networks. Mentions both the general principles of spatial databases and their implementation on a specific technology platform used in this study as a modeling tool. Proposed implementation on this platform of localization algorithm described in chapter four. Outlined principles of simulation models construction and scenarios of the algorithm. In conclusion, discusses the results of processing and analysis of the data.

Localization of automated vehicles using digital maps as an additional information resource is the most promising direction for solving the localization problem. A special role is played here by GIS system serving a useful tool for bidirectional conversion of symbolic geography and global *(x, y)* addresses.

GIS can also be extremely useful in the landmark pattern recognition, location and properties of which are usually stored in a databases [47]. The only restriction for this kind of localization is the presence of at least one reference point, or at least one GNSS receiver that allows you to bind a localization process to specific GIS context. In the latter case, high precision GNSS receiver is not a critical parameter. In addition, good practices can be a preliminary stage of training for GIS data collection and landmark characteristics refinement, if such recognition supported by a localization system.

## 5.1 Spatial queries

Based on location spatial requests are related to the type of requests, the result of which depends, also on a location of the requesting client. Effectiveness of management and processing of spatial queries becomes critical with the development and growth of wireless and mobile technologies. Such requests, in contrast to the traditional methods of querying databases, have certain unique features.

In particular, there has been a marked increase of interest [146] to the spatial queries to information based on the user's current location of the mobile network. Among wireless networks, communication constraints also are important in choosing the spatial query processing strategies. In this context, the implementation of localization algorithm based on spatial queries assumes that the network user has the ability to establish a direct connection with the information server and receive answers to their queries on demand.

As one of the possible implementation of approach to this problem Oracle Spatial 11g package was chosen. This well-known and well-proven software package contains a database and all the necessary functions and procedures that allow you to store a wide range of spatial data, as well as to organize a quick and efficient access to data and their analysis. Spatial data are in this case, the main characteristics of the location of the real or conceptual objects and their relationships in the real or conceptual space of their existence.

Oracle Spatial 11g [147] provides the user with SQL schema and functions that provide storage, retrieving and updating sets of spatial elements, as well as requests to database platform Oracle 11g. Spatial database platform consists of the following main components:

- Schemes (MDSYS), which determines the order of storage, the syntax and semantics of supported geometric data types;
- Mechanism of spatial indexing;
- Operators, functions, and procedures for performing queries in a designated location, spatial connection requests and other analytical operations;
- Functions and procedures for debugging and tuning;
- Topological data model;
- Network data model;
- GeoRaster, functionality allowing storing, index, query, and analyze raster data and associated metadata.

The spatial component of the considered data structures, hereinafter referred to by the term “geometry” is a geometrical representation of the object shape in a certain coordinate space. Spatial database typically maintains object-oriented relationship for geometries. For these purposes, the Oracle platform provides data model SDO_GEOMETRY, which in vector form stores the description of geometric characteristics of the object and the metadata associated with the object. Oracle table can contain one or more columns of type SDO_GEOMETRY. The implementation of an object-oriented data model fully complies with the “SQL with geometric types” open standard for geolocation systems Open GIS ODBC / SQL.

Let us consider a road map as the most common example of spatial data. Road map is a two-dimensional object containing the geometry types such as point, line and polygon, which in turn represents cities, roads, administrative and political boundaries. Essentially, it is a geographical information visualization. Actual location of cities, roads, and political boundaries at the surface, projected onto the two-dimensional plane while maintaining the relative distance of objects displayed in the map.

Applications designed to work with these types of data, typically support storage, retrieval, modification and queries of data sets having both spatial and other attributes. Examples of non-spatial attributes one can mention a soil type, land use classification, parcel number and so on. Spatial attributes are coordinates of geometries, vector representation of the forms or other object properties.

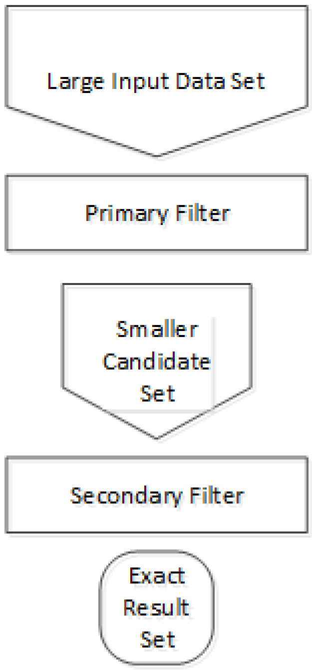



***Fig. 34.*** *Spatial query model*

The method of obtaining the compound or spatial data uses a model consisting of two queries. Such model design means two independent successive operations performed to obtain the results of the spatial query. The combination of the results of each of the two operations is the final result of the query. As shown in Fig. 34, the first operation acts as a filter for a large amount of data. It significantly reduces the number of candidate solutions for their further processing. The second, more accurate filtering operation works with a small amount of data and returns the final result of the query.

The possibility of spatial database indexing is one of the key new features in Oracle Spatial product. A spatial index, as well as any other index, provides a mechanism to restrict the search window. In this case, the limitation is based on the spatial criteria such as geometries intersection, overlapping, etc. This functionality provides the use of R-tree indexing (Fig. 35).

R-tree, a tree-like data structure that is used for to access spatial data, i.e., for multi-dimensional information indexing, such as geographic data with two-dimensional coordinates. This data structure divides the space into a set of hierarchically nested and possibly overlapping rectangles (two-dimensional space). In the case of three-dimensional or multidimensional space, it will be a set of rectangular parallelepipeds (cuboids) [148].

Insertion and removal algorithms of these bounding rectangles used to ensure that the "closely spaced" objects are placed in one leaf level. In particular, the new object gets into the leaf level, which would require the smallest expansion of its bounding box. Each element of the leaf top keeps two data fields: a method for identifying data describing the object (or the data itself) and the bounding rectangle of the object.

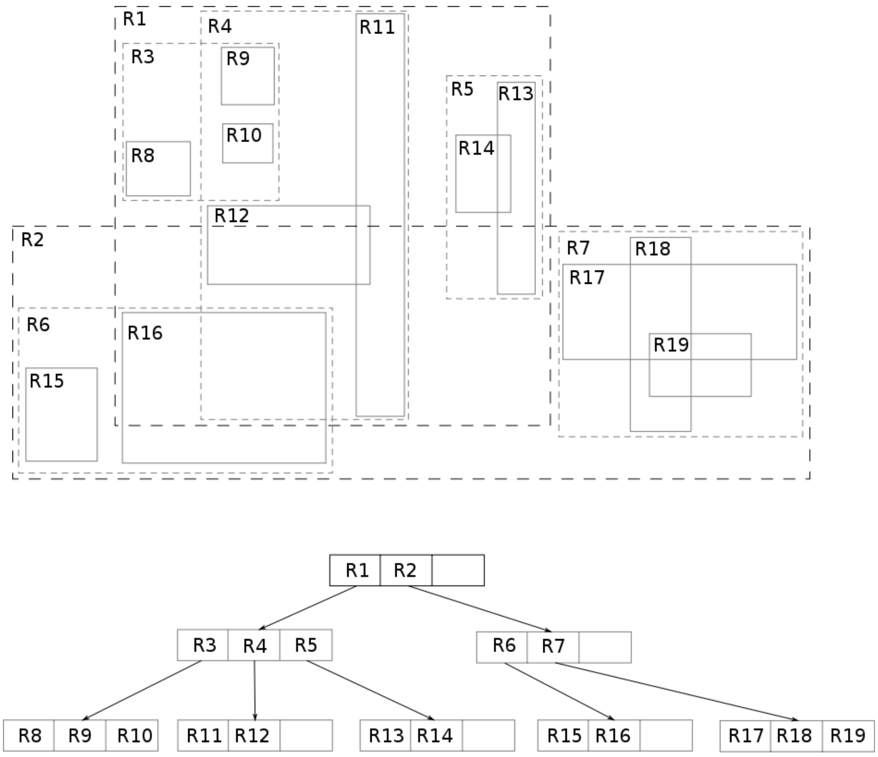



***Fig. 35.*** *R-tree for two-dimensional rectangles*

Similarly, insertion and deletion algorithms, search algorithms (e.g., intersection, inclusion, and neighborhood) use the bounding rectangles for a decision on the need to find a child node. Thus, most vertices are not affected by the search. This property of R-trees makes them applicable to the database where the vertices can be unloaded to the disc on demand. For splitting overcrowded vertices different algorithms can be used that generates division R-trees into subtypes: quadratic and linear.

Initially indexes based on R-tree worked quite well on real data, but good results for worst-case scenarios were not guaranteed. In 2004 was published a new, improved algorithm [149], which determines the priority R-trees. An improved algorithm not only showed the results at the level of the most effective modern methods, but its optimality for worst-case scenarios.

### 5.2 Proposed method

Computational complexity of the localization algorithm, proposed in the fourth chapter of this research, can reach unacceptably high values in various scenarios, in some cases up to *O(N!).* In this connection, it seems appropriate to test its implementation as a method of spatial queries, which should allow largely optimizing the performance of the algorithm and reducing the influence of the localization scenario on requirements to computing resources.

Using spatial query functionality described above, consider the implementation of the algorithms mentioned in the form of spatial SQL operations. For the implementation of the algorithm two tables will be required, each of which has a column of type SDO_GEOMETRY.

The tables use a primary key and record type identifiers. Other columns are system and are not directly related to the performed algorithm logic.

The first table AREA_MAP designed to store digital maps, including an anchor point *b* with known coordinates as required by the considered network localization scenario. Furthermore, during scenario processing in the same table are recorded not localized nodes *1* ... *n*, then for those whose distance from the anchor node is taken as known, constructed set of circles *C*, with a center having a coordinate $(b_x, b_y)$ and radius $r_i$ equal to the distance to the node *i*, the distance $d_i$ to which was assumed as known.

The second table STAGING_MAP, is the staging table, needed to store temporarily the intermediate results of the algorithm. Spatial indexing procedure not described here specifically, since it is not part of the localization algorithm logic. However, for spatial queries high performance, spatial indexes update must be performed after any changes in the digital map or the location of the anchors.

The whole localization process is performed by two SQL operations. The first operation (Table 9), calculates all the points of intersection of the circles *C* and digital map geometries elements, and stores these points in the staging table with a mark indicating to which of the circles $c_i$ each given point belongs.

The operation uses a primary filter ANYINTERACT (any geometric interaction), to narrow the search window, and then function SDO_INTERSECTION, which have primary responsibility for the calculation of the total number of points of intersection. Last, the third parameter of the function, map tolerance, in this case set to value of 0.005. Description and examples of the filter functions ANYINTERACT and SDO_INTERSECTION usage are given in Appendix N2.

The second stage of the localization algorithm, SQL operation 2 (Table 10), is a selection of the set of unique points of intersection from the staging table, satisfying conditions of the corresponding matrix of known distances, both between nodes and between nodes and anchor.

In Table 10, parameter $L_n$ stands for measured distance between nodes $n_i$ and $n_{i+1}$ and $D_n$ - the distance between the anchor *b* and the node $n_i$. Tolerance in this case is not defined by a constant, but is passed as a parameter of the operation that allows simulating different localization scenarios.

In the case where the graph formed by nodes, is rigid enough or additional information obtained by means of a digital map, makes it rigid enough, the result of the query will be a unique set of points corresponding to the real topological arrangement of network nodes. However, when information is insufficient, multiple localization results are possible, one of which is real and the other will be products of rotation in the case of one anchor, or products of reflection in the case of the two anchors.

**Table 9.** SQL Operation 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';
```

***Table 10.*** *SQL Operation 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;
```

The results of the localization algorithm in different scenarios now will be considered in more detail.

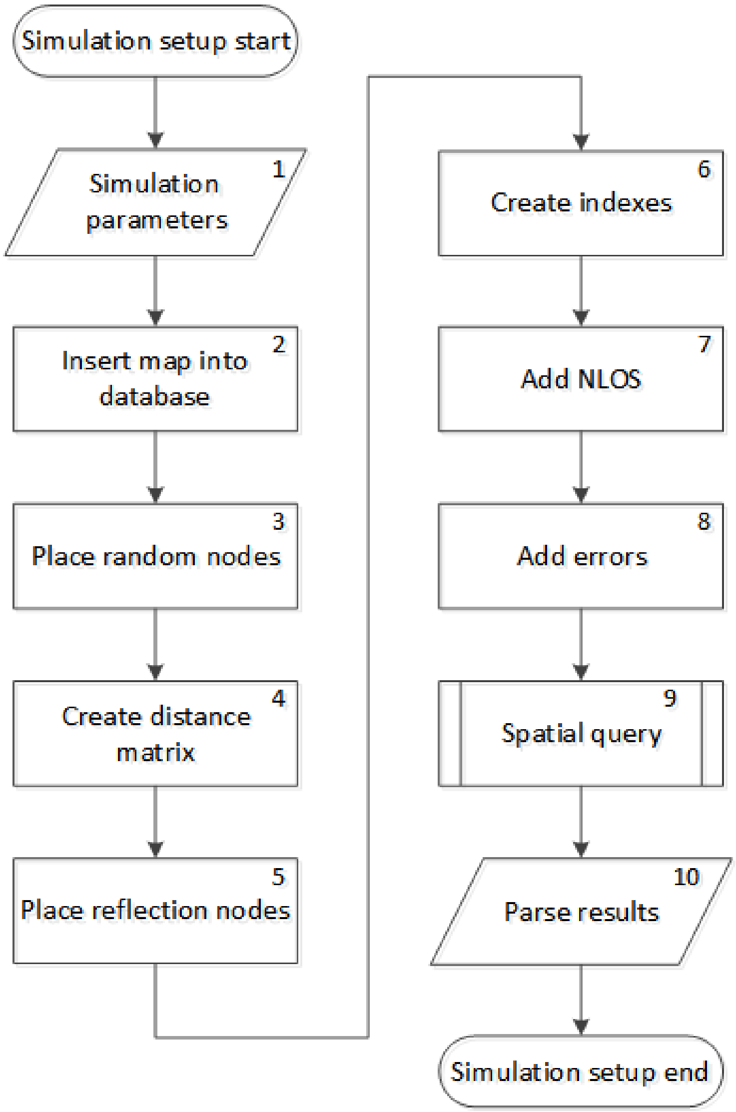



***Fig. 36.*** *Simulation algorithm*

### 5.3 Simulation environment

As previously, the modeling is performed using a discrete simulation model to perform testing of the possible ways of events development, and the behavior of the object by variation of some or all of the model parameters. Furthermore, the model can be described as geometrically graphic, as it is presented largely by geometric objects and images appropriately coded in spatial database.

In this model, an algorithm includes a simulation environment preparation and setting up the scenario. Algorithm block diagram shown in Fig. 36.

The task of blocks 1 to 8 is to prepare digital maps, placement of network nodes according to the selected scenario and setting of simulation configuration parameters and indices. Localization is performed by spatial queries in block 9. The objective of the latter, the

10th block is processing the results and preparation of data for visualization. The main classes of localization applications are given in Appendix N3.

### 5.4 Spatial query profiling

An important stage in the implementation of both spatial and conventional SQL queries is their profiling, collection characteristics of the program in order of further optimization that includes the search for an optimal query execution plan of all possible for a given query and process of query or the database structure changes to reduce usage of computing resources while executing the query. The same result can be obtained DBMS by in various ways (the query plan), which may significantly differ on a cost of resources and execution time.

In the implementation of the model on the Oracle Spatial platform, the built-in Tuning Optimizer was used for profiling and debugging. The resulting report of this tool is performed detailed analysis of requests and recommendations for its improvement, followed by rational justification for the proposed steps. Usually, the recommendation suggests establishment of certain indexes, restructuring requests and provides detailed profiling results of the request.

Recommendations for improvement schemes and analysis of performance of the original spatial queries are presented in Table 11 and Table 12. The plan of the query contains 10 nested program cycles and 11 accesses to a full reading of the table containing a digital map. Each step of the query plan execution time is insignificant and showed as a minimum value for the data format of the results time table. Data size is small as well, which is typical for vector geometry data type, and is only 330 bytes. At the same time, CPU usage when running a query reaches 55%.

In this regard, it is proposed to optimize the structure of the database that is to introduce additional composite index by type of geometry of a digital map. Improvements rationale shown in Table 13, the results of the optimized spatial query are shown in Table 14.

***Table 11.*** *Improvement recommendations*

```
Recommendation (estimated benefit: 54.59%)
------------------------------------------
Consider running the Access Advisor to improve the physical schema design
or creating the recommended index.
create index EXPERT.IDX$$_13620001 on
EXPERT.AREA_MAP("SEGMENT_NAME","SEGMENT_ID");
```

***Table 12.*** *Original query plan*

| Id | | Operation | Name | Rows | Bytes | Cost(%CPU) | Time |
|---|---|---|---|---|---|---|---|
| | 0 | SELECT STATEMENT | | 1 | 330 | 55 (0) | 00:00:01 |
| | 1 | NESTED LOOPS | | 1 | 330 | 55 (0) | 00:00:01 |
| | 2 | NESTED LOOPS | | 1 | 300 | 50 (0) | 00:00:01 |
| | 3 | NESTED LOOPS | | 1 | 270 | 45 (0) | 00:00:01 |
| | 4 | NESTED LOOPS | | 1 | 240 | 40 (0) | 00:00:01 |
| | 5 | NESTED LOOPS | | 1 | 210 | 35 (0) | 00:00:01 |
| | 6 | NESTED LOOPS | | 1 | 180 | 30 (0) | 00:00:01 |
| | 7 | NESTED LOOPS | | 1 | 150 | 25 (0) | 00:00:01 |
| | 8 | NESTED LOOPS | | 1 | 120 | 20 (0) | 00:00:01 |
| | 9 | NESTED LOOPS | | 1 | 90 | 15 (0) | 00:00:01 |
| | 10 | NESTED LOOPS | | 1 | 60 | 10 (0) | 00:00:01 |
| * | 11 | TABLE ACCESS FULL | AREA_MAP | 1 | 30 | 5 (0) | 00:00:01 |
| * | 12 | TABLE ACCESS FULL | AREA_MAP | 1 | 30 | 5 (0) | 00:00:01 |
| * | 13 | TABLE ACCESS FULL | AREA_MAP | 1 | 30 | 5 (0) | 00:00:01 |
| * | 14 | TABLE ACCESS FULL | AREA_MAP | 1 | 30 | 5 (0) | 00:00:01 |
| * | 15 | TABLE ACCESS FULL | AREA_MAP | 1 | 30 | 5 (0) | 00:00:01 |
| * | 16 | TABLE ACCESS FULL | AREA_MAP | 1 | 30 | 5 (0) | 00:00:01 |
| * | 17 | TABLE ACCESS FULL | AREA_MAP | 1 | 30 | 5 (0) | 00:00:01 |
| * | 18 | TABLE ACCESS FULL | AREA_MAP | 1 | 30 | 5 (0) | 00:00:01 |
| * | 19 | TABLE ACCESS FULL | AREA_MAP | 1 | 30 | 5 (0) | 00:00:01 |
| * | 20 | TABLE ACCESS FULL | AREA_MAP | 1 | 30 | 5 (0) | 00:00:01 |
| * | 21 | TABLE ACCESS FULL | AREA_MAP | 1 | 30 | 5 (0) | 00:00:01 |

Now in the query plan there was only one table full access to digital maps and 10 index-based accesses to the type of map element geometry. As it can be seen, result of CPU utilization decreased down to 25%. Therefore, to support localization implemented by proposed method, twice as less computing power will be needed, what is a good profiling result.

*Table 13. Scheme improvement rationale*

```
Rationale
---------
Creating the recommended indices significantly improves the execution plan
of this statement. However, it might be preferable to run "Access Advisor"
using a representative SQL workload as opposed to a single statement. This
will allow to get comprehensive index recommendations which takes into
account index maintenance overhead and additional space consumption.
```

*Table 14. Query plan with additional indexes*

| | Id | Operation | Name | Rows | Bytes | Cost(%CPU) | Time |
|---|---|---|---|---|---|---|---|
| | 0 | SELECT STATEMENT | | 1 | 330 | 25 (0) | 00:00:01 |
| | 1 | NESTED LOOPS | | 1 | 330 | 25 (0) | 00:00:01 |
| | 2 | NESTED LOOPS | | 1 | 300 | 23 (0) | 00:00:01 |
| | 3 | NESTED LOOPS | | 1 | 270 | 21 (0) | 00:00:01 |
| | 4 | NESTED LOOPS | | 1 | 240 | 19 (0) | 00:00:01 |
| | 5 | NESTED LOOPS | | 1 | 210 | 17 (0) | 00:00:01 |
| | 6 | NESTED LOOPS | | 1 | 180 | 15 (0) | 00:00:01 |
| | 7 | NESTED LOOPS | | 1 | 150 | 13 (0) | 00:00:01 |
| | 8 | NESTED LOOPS | | 1 | 120 | 11 (0) | 00:00:01 |
| | 9 | NESTED LOOPS | | 1 | 90 | 9 (0) | 00:00:01 |
| | 10 | NESTED LOOPS | | 1 | 60 | 7 (0) | 00:00:01 |
| * | 11 | TABLE ACCESS BY INDEX ROWID | AREA_MAP | 1 | 30 | 2 (0) | 00:00:01 |
| * | 12 | INDEX RANGE SCAN | IDX$$_13 | 1 | | 1 (0) | 00:00:01 |
| * | 13 | TABLE ACCESS FULL | AREA_MAP | 1 | 30 | 5 (0) | 00:00:01 |
| * | 14 | TABLE ACCESS BY INDEX ROWID | AREA_MAP | 1 | 30 | 2 (0) | 00:00:01 |
| * | 15 | INDEX RANGE SCAN | IDX$$_13 | 1 | | 1 (0) | 00:00:01 |
| * | 16 | TABLE ACCESS BY INDEX ROWID | AREA_MAP | 1 | 30 | 2 (0) | 00:00:01 |

*Table 14.* Continuation

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| * | 17 | INDEX RANGE SCAN | IDX$$_13 | 1 | | 1 (0) | 00:00:01 |
| * | 18 | TABLE ACCESS BY INDEX ROWID | AREA_MAP | 1 | 30 | 2 (0) | 00:00:01 |
| * | 19 | INDEX RANGE SCAN | IDX$$_13 | 1 | | 1 (0) | 00:00:01 |
| * | 20 | TABLE ACCESS BY INDEX ROWID | AREA_MAP | 1 | 30 | 2 (0) | 00:00:01 |
| * | 21 | INDEX RANGE SCAN | IDX$$_13 | 1 | | 1 (0) | 00:00:01 |
| * | 22 | TABLE ACCESS BY INDEX ROWID | AREA_MAP | 1 | 30 | 2 (0) | 00:00:01 |
| * | 23 | INDEX RANGE SCAN | IDX$$_13 | 1 | | 1 (0) | 00:00:01 |
| * | 24 | TABLE ACCESS BY INDEX ROWID | AREA_MAP | 1 | 30 | 2 (0) | 00:00:01 |
| * | 25 | INDEX RANGE SCAN | IDX$$_13 | 1 | | 1 (0) | 00:00:01 |
| * | 26 | TABLE ACCESS BY INDEX ROWID | AREA_MAP | 1 | 30 | 2 (0) | 00:00:01 |
| * | 27 | INDEX RANGE SCAN | IDX$$_13 | 1 | | 1 (0) | 00:00:01 |
| * | 28 | TABLE ACCESS BY INDEX ROWID | AREA_MAP | 1 | 30 | 2 (0) | 00:00:01 |
| * | 29 | INDEX RANGE SCAN | IDX$$_13 | 1 | | 1 (0) | 00:00:01 |
| * | 30 | TABLE ACCESS BY INDEX ROWID | AREA_MAP | 1 | 30 | 2 (0) | 00:00:01 |
| * | 31 | INDEX RANGE SCAN | IDX$$_13 | 1 | | 1 (0) | 00:00:01 |

### 5.5 Scenario evaluation

The purpose of the simulation is to evaluate the performance of the proposed algorithm under different network sizes on different configurations of digital maps, different visibility conditions, as between the nodes, and the visibility of the anchor, as well as to evaluate the stability of the algorithm to the distance measurement errors.

For the simulation of the localization algorithm were chosen two random fragment of typical urban road infrastructure [150]. Both fragments were transferred to database as composite spatial geometries and hereinafter referred are as digital map F (Fig. 37) and G (Fig. 38). It is worth mentioning that as a map F area with a high degree of symmetry was deliberately selected and which in localization process should provide a high probability of errors like rotation, reflection, and flex [151]. Map is G more asymmetric and belongs to the optimistic scenario simulation.

Evaluation starts with generating a predetermined number of nodes with a uniform distribution of random coordinate within geometric elements of the map. One random node is chosen as an anchor with a priori known coordinates. It is assumed that each node has a line of sight, and thus the measured distances to a given number of neighbor nodes. The remaining nodes are not in line of sight (NLOS). Also, specified the number of nodes with NLOS to the anchor node. By default, the distance to the anchor node is known. To emulate the distance measurement error, to the tolerance parameter added normally distributed random variable with expectation 0 and variance given, which if necessary can distort the results of the request within the predefined limits.

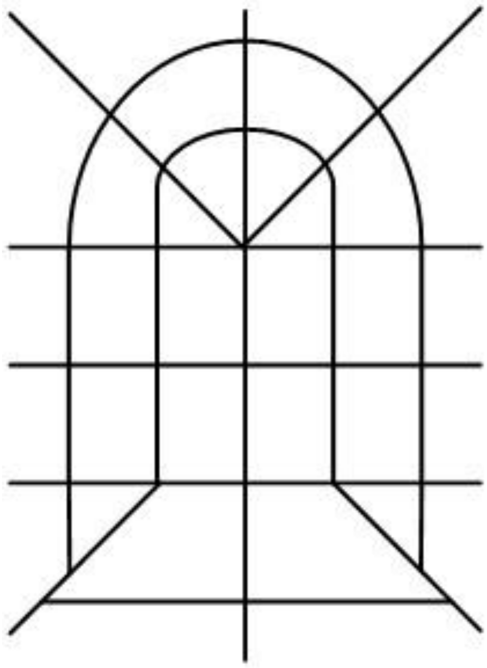


***Fig. 37.*** *Map F, symmetric*

For each of the map different simulation scenarios have chosen, which consider the possibility of successful localization for networks of different sizes and different degrees of the graph connectivity, which is reflected in the proportion of nodes with NLOS. Scenarios parameters discussed in more detail below. At the end of each stage of the simulation, program returns detailed data log. For clarity and ease of understanding, the results are processed and presented in the form of three-dimensional surface diagrams.

Each chart shows the results of localization depending on the scenario parameters. This can be uniquely localized network, a number of possible placement solutions or absence of results in the case where no solution is found or the number of possible placements is too high and exceeds a predetermined threshold value.

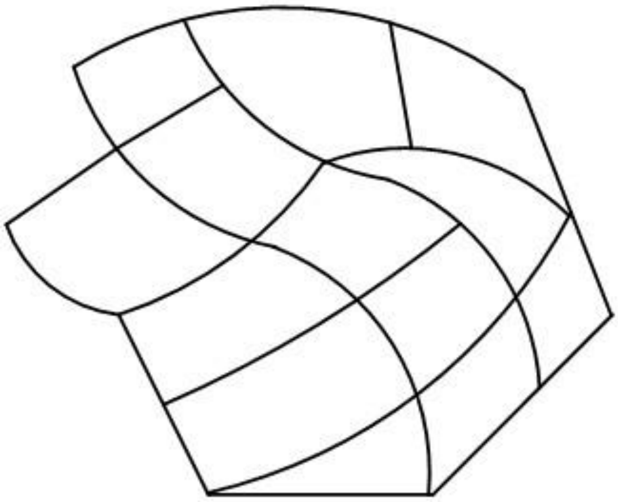


***Fig. 38.*** *Map G, organic*

The number of solutions shown on the axis Y, in the range from 0 to 50. For practical reasons, the number of solutions more than 50, has been considered as not determined and taken as a failure. NLOS percentage ratio between network nodes shown on the axis X, and NLOS percentage ratio between nodes and anchor on the axis Z.

### 5.6 Experiment data

Experimental verification of the proposed algorithm included wireless network scenarios localization simulation using spatial queries, and consisted of three main stages. The objective of the first step was to explore in general form of network algorithm performance in a wide range of parameter values as well as the size of the network and nodes visibility conditions NLOS for various localization scenarios. The result should become the initial data on the impact of the scenarios and the critical region of the parameters of the environment when the quality of the algorithm begins to deteriorate and further localization becomes impossible.

The second phase of the study is aimed at a more detailed study of a narrow range of the environment parameters values to obtain more accurate data on the properties and the most critical parameters of the algorithm. The purpose of the third stage was to simulate different measurement errors values for the algorithm provided in the critical range that gives an opportunity to evaluate the order of the measuring accuracy requirements to network hardware.

Example parameters of one iteration of the simulation are given in Table 15. It is shown that the simulation is carried out on two maps for the three networks in sizes 5, 25 and 45 nodes. Each of the networks is placed randomly three times. For each placement localization algorithm is executed under the conditions of 70, 75, 85 and 90 per cent of random NLOS, so as for nodes and for anchors.

***Table 15.*** *Simulation plan sample*

| Random placements | Nodes | Map 1 | | | Map 2 | | |
|---|---|---|---|---|---|---|---|
| | **NLOS%** | 5 | 25 | 45 | 5 | 25 | 45 |
| | 3 | 70 | 70 | 70 | 70 | 70 | 70 |
| | 3 | 75 | 75 | 75 | 75 | 75 | 75 |
| | 3 | 85 | 85 | 85 | 85 | 85 | 85 |
| | 3 | 90 | 90 | 90 | 90 | 90 | 90 |

For the first localization scenario map G was selected. Percent of nodes that are unable to measure the distance, and hence having NLOS with the anchor or NLOS with neighboring nodes, ranging from 70% to 90% of the total number of nodes, incrementing of 5% each iteration of the simulation. Furthermore, the localization scenario was repeated for networks of 5, 25 and 45 nodes. Thus, a simulation of the selected placement on the map consisted of 75 localization scenarios runs. Totally 3 random placement of nodes and anchors on the map was made, so the total number of runs in one simulation algorithm was equal 225.

The results of the simulation for networks consisting of 25 and 45 nodes showed 100% successful localization for all selected NLOS values. Non-unique results emerged only in the network with the fewest nodes. In Fig. 39 is given a three-dimensional graph of the first placement on the map G network consisting of five nodes.

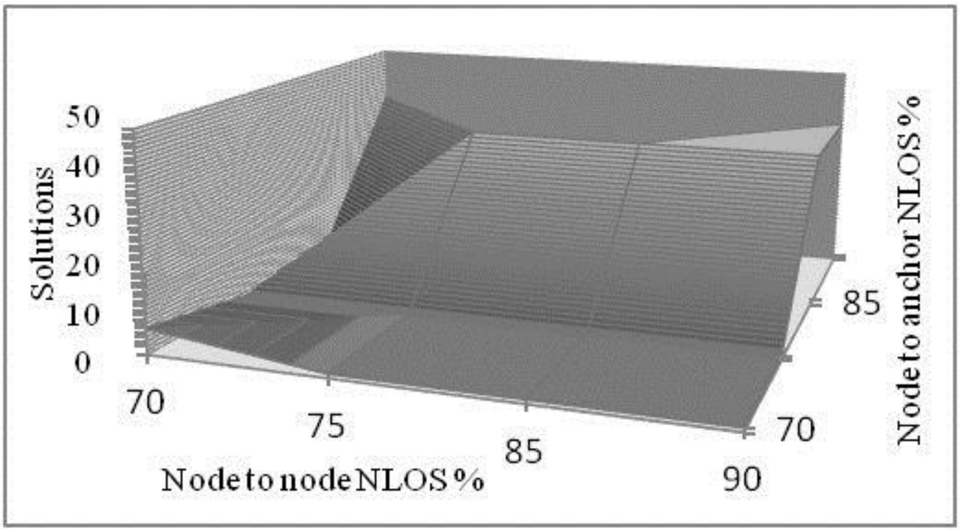



***Fig. 39.*** *Map G, 5 nodes, placement 1*

Similar results were obtained in two other random placements. Networks of 25 and 45 nodes showed only the unique successful localization results. Localization of network consisting of five nodes, in some cases, showed some non-unique results as shown in Fig. 40 and Fig. 41.

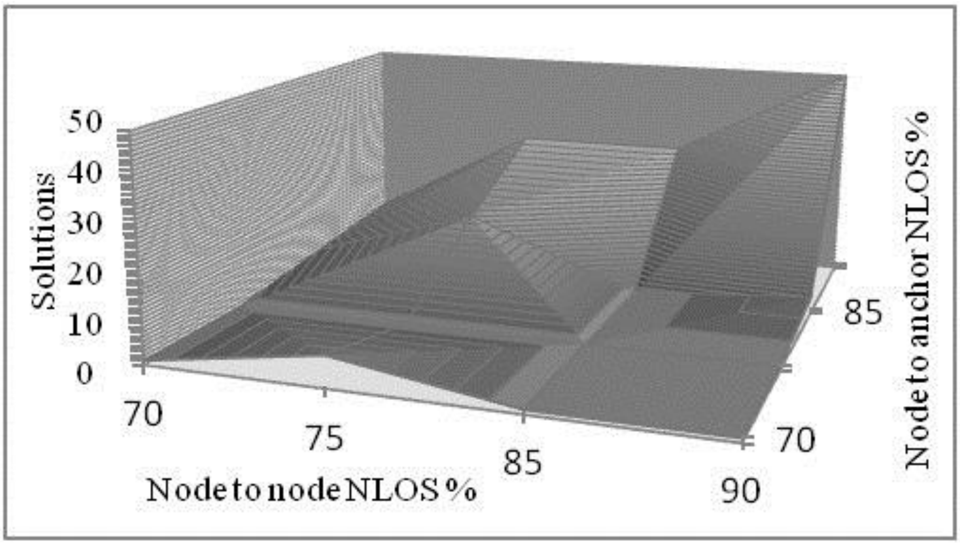



***Fig. 40.*** *Map G, 5 nodes, placement 2*

As can be seen from the above diagrams greatest, stabile localization, up to 90% NLOS, is observed in poor visibility between nodes. Although intuitively it is assumed that a larger amount of data must always improve the results, one can see here that on the contrary, the excess data in some cases adds from two to five alternative placements. In turn, a large number of NLOS from node to the anchor generates up to 50 alternative placements, because of the high degree of map G symmetry and thus the possibility of transformation of rotation.

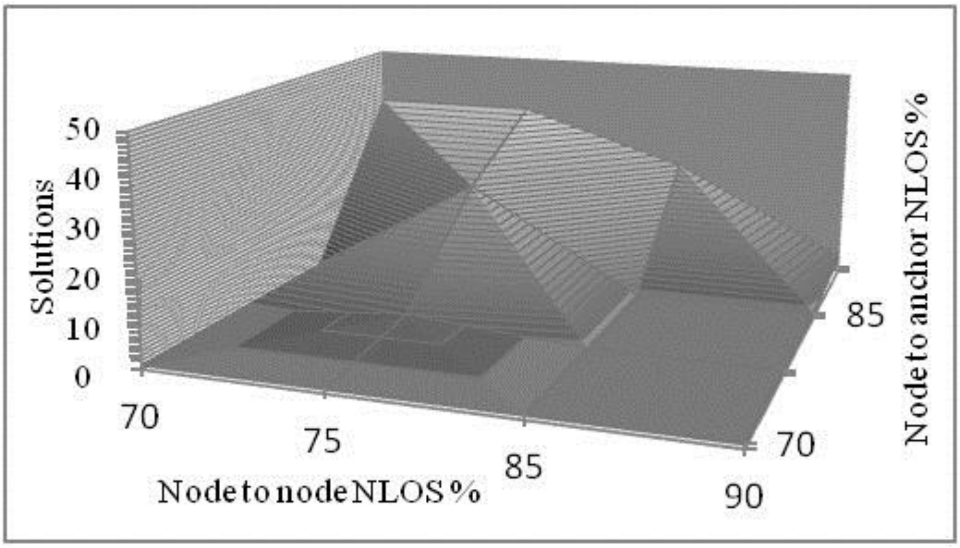



***Fig. 41.*** *Map G, 5 nodes, placement 3*

Next, simulation was repeated for the same scenarios but the map F, having more asymmetric properties. Similarly, stable results with 100% successful localization were obtained for networks of 25 and 45 nodes. A network of 5 nodes at large NLOS degree shows alternative non-unique placements. All three placements are shown in Figures 42, 43 and 44.

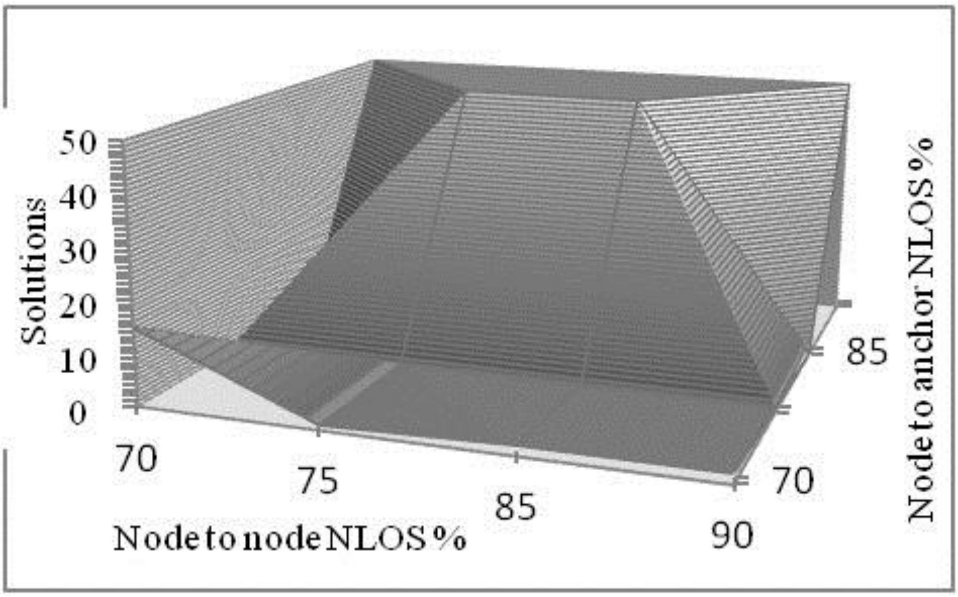



***Fig. 42.*** *Map F, 5 nodes, placement 1*

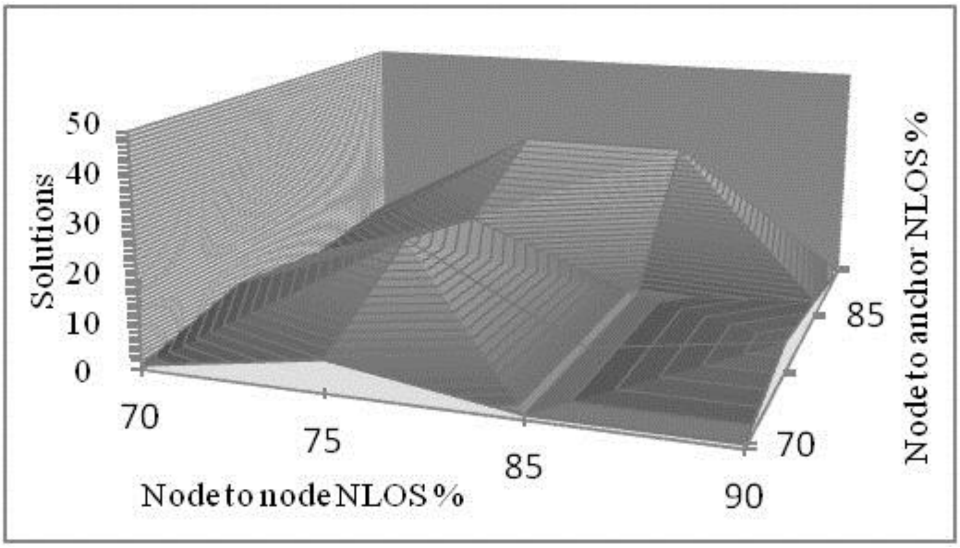



***Fig. 43.*** *Map F, 5 nodes, placement 2*

It should be noted that the range and number of non-unique placements for large values of NLOS in this scenario have increased. This illustrates the greater network graph susceptibility to transformations such as reflection and flex what is characteristic for the map with a lower degree of symmetry. However, none of the scenarios has not gone beyond the predefined maximum of non-unique solutions, when the result of localization is treated as unsuccessful.

Previous simulation showed that when the size of the network exceeds 20 nodes and the number of nodes without a line of sight NLOS is up to 80% of the total number of nodes, algorithm shows stable successful results.

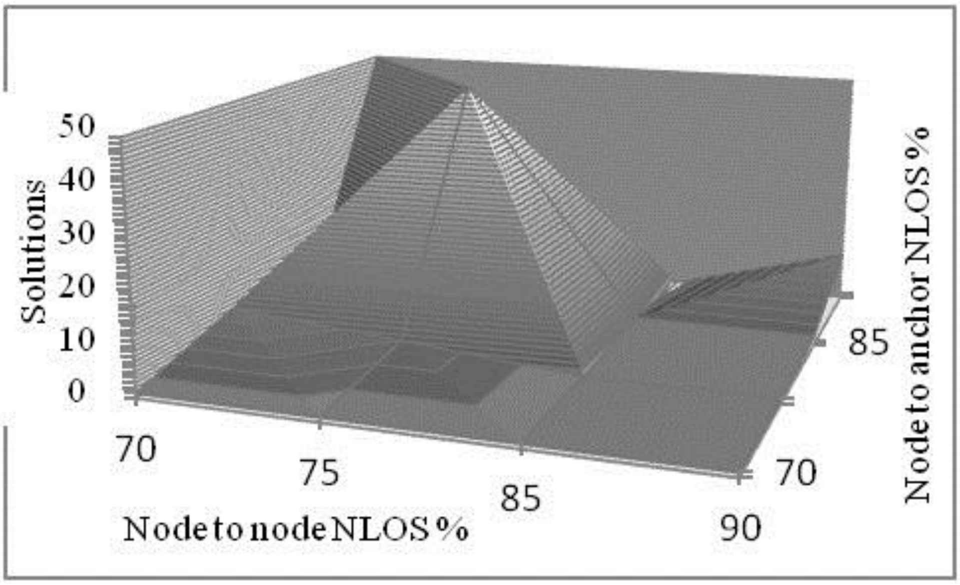



***Fig. 44.*** *Map F, 5 nodes, placement 3*

Therefore, in further simulations chosen NLOS range will be from 85% to 97% and the size of the network will be decreased from 30 nodes to 5 in decrements of 5 nodes. This allows getting a more qualitative view to analyze and plot parameters of the algorithm on the interval when it stops working. Moreover, since the previous simulation results showed insignificant dependence on the type of map and the number of random placements, further simulations were carried out on the map G with one placement.

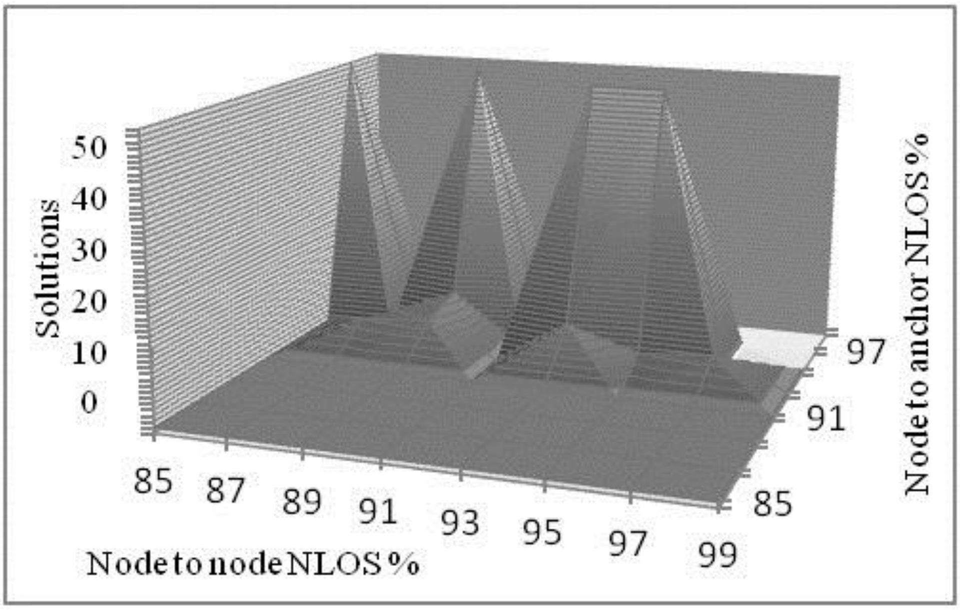



***Fig. 45.*** *Map G, 30 nodes*

Diagrams presented in Fig. 45 to 50 shows the results of simulations carried out with six parameters mentioned above.

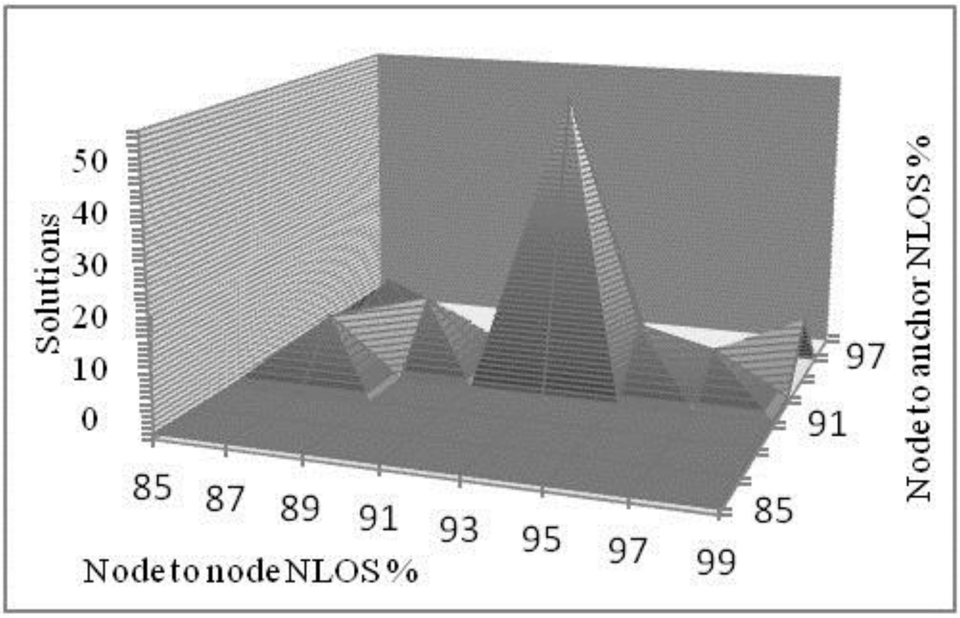



***Fig. 46.*** *Map G, 25 nodes*

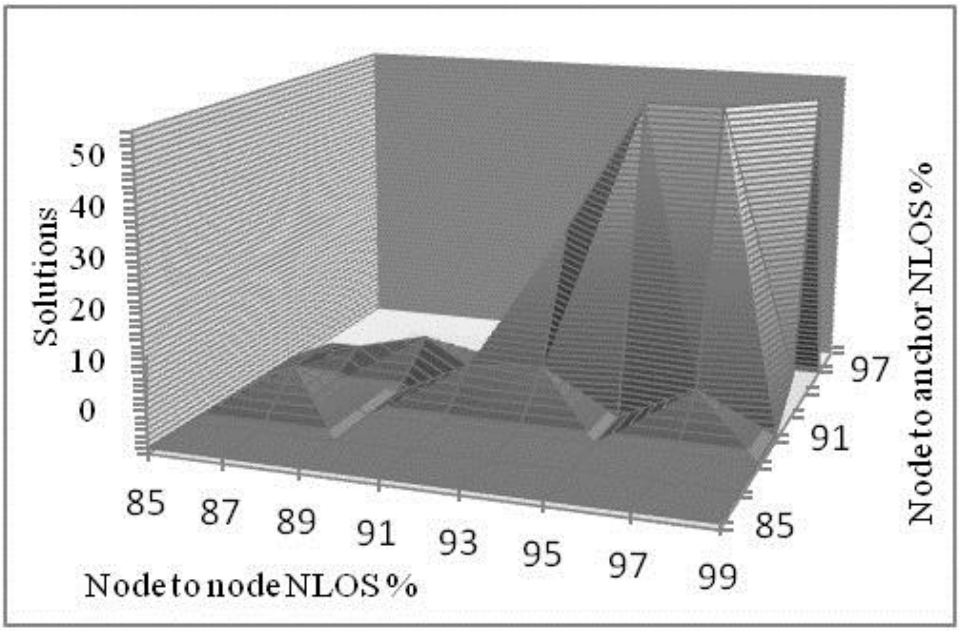



***Fig. 47.*** *Map G, 20 nodes*

Localization scenario of network consisting of 30 nodes, as shown in Fig. 45, shows a stable and successful localization and unique results over the entire range of selected NLOS size from node to node, provided that the NLOS from nodes to the anchor does not exceed 90%. Moreover, at the maximum NLOS from node to node, when NLOS from the nodes to anchor, exceeding 95%, the first undefined results occurred (Fig. 46). This means that the limit is reached when the number of possible alternative placements becomes computationally difficult and localization algorithm stops operating. While the size of the network decreases, as shown in Figures 47, 48 and 49, deterioration of localization results progresses.

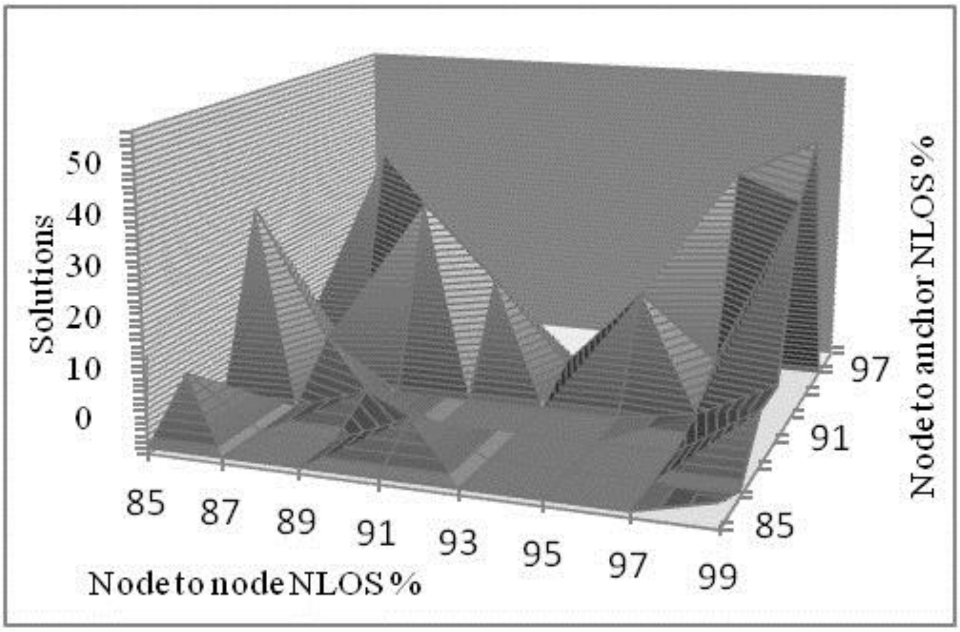



***Fig. 48.*** *Map G, 15 nodes*

Last, the most pessimistic scenario with a network size of 5 nodes and extremely low degree of connectivity of the graph, the results of the localization of which are shown in Fig. 50, does not result in a single unique location. The whole area beyond 90% of NLOS from nodes to the anchor is not defined. However, with such a small network size, 90% NLOS would mean that only one node has information about the anchor and its range what it is insufficient when the network configuration graph is poorly connected.

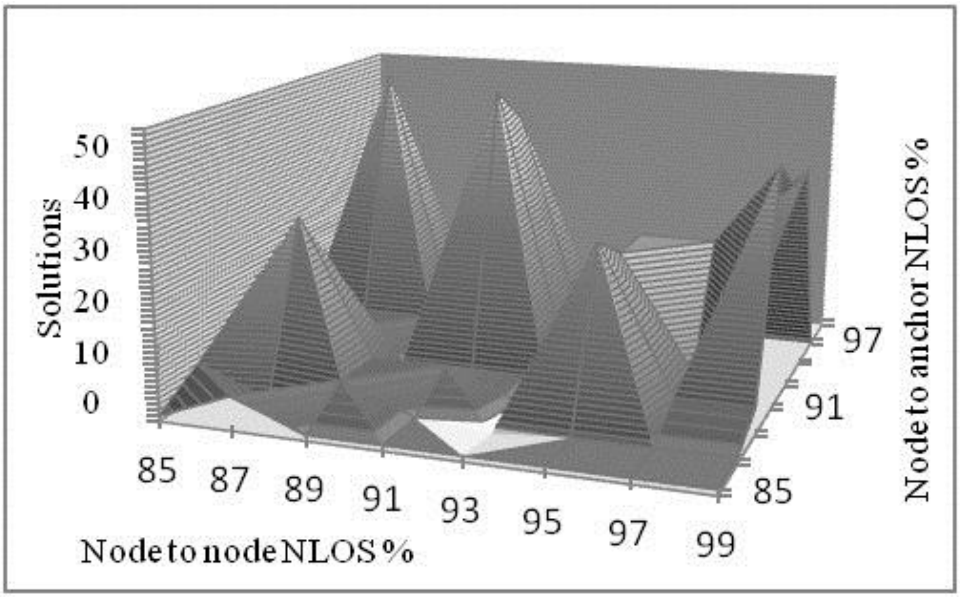



***Fig. 49.*** *Map G, 10 nodes*

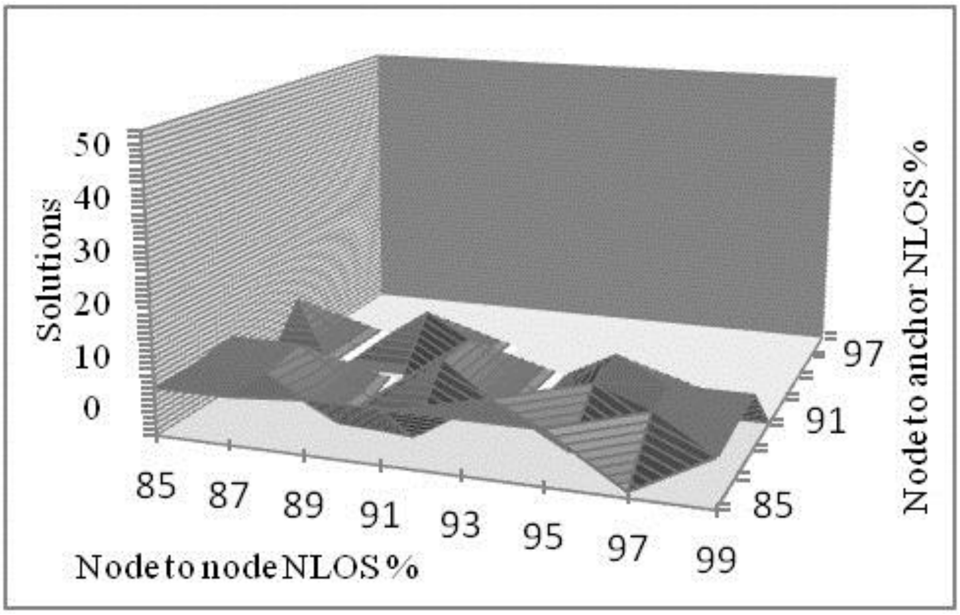



***Fig. 50.*** *Map G, 5 nodes*

The last three scenarios were intended to simulate and evaluate overall impact of the measurement error magnitude. Previously, the default measurement error was set at 0.05m which corresponds to the boundary of statistical error, and does not significantly affect the results. In further simulations error size was set as normally distributed random variable with zero mean $\mu = 0$ and standard deviation $\sigma = 1$ for the first scenario, $\mu = 1$, $\sigma = 5$ for the second and $\mu = 5$, $\sigma = 25$ for the third scenario respectively. This means that the magnitude of the measurement error can reach 25 meters.

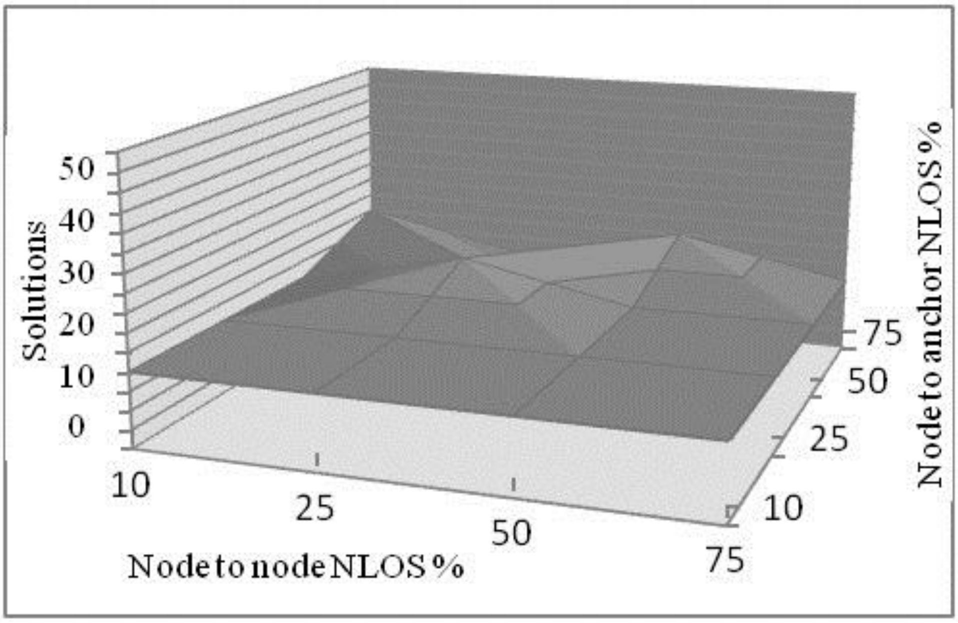



***Fig. 51.*** *Map G, 20 nodes, error 0m – 1m*

It should be noted that at the largest allowable error of 25 meters, the localization algorithm functioning was obviously impossible. Since the error limit of 25 meters is comparable to the placed network size, the size of the geometric component of a digital map and the distance between them in the context of urban development, this method of localization becomes disoriented and gives no results. Therefore, in further we consider only the first two scenarios parameters $\mu = 0$, $\sigma = 1$ and $\mu = 1$, $\sigma = 5$.

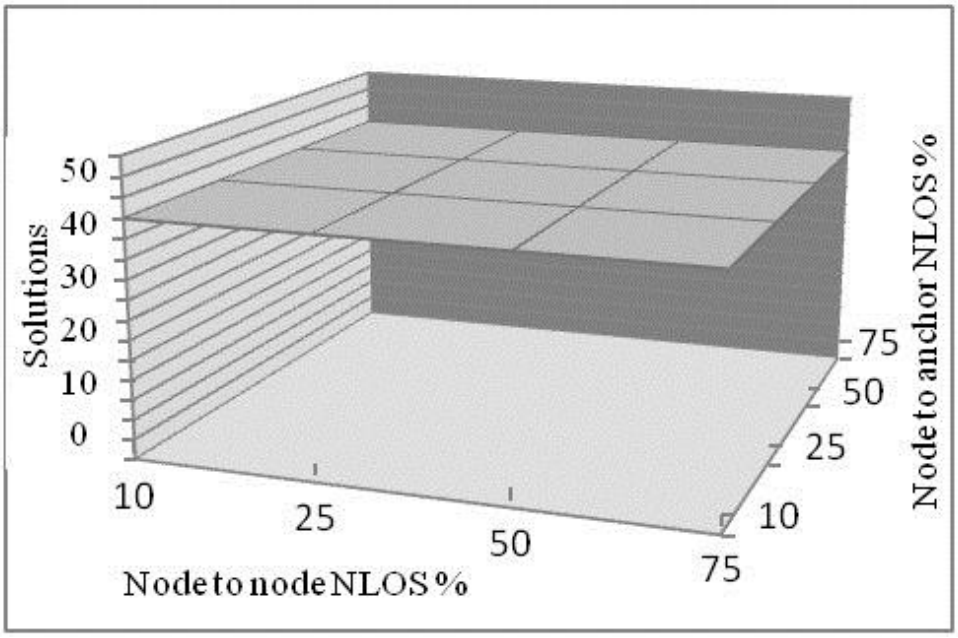



***Fig. 52.*** *Map G, 20 nodes, error 1m – 5m*

Simulation results of the last two scenarios are reflected in the Figures 51 and 52. Measurement errors are implemented with tolerance parameter of spatial query, what in turn erodes the value of the measured distances. In both cases, settings used are the most approximate to the real. Studied network has size of 20 nodes and NLOS values ranged from 10% to 75%. The results showed that the measurement error of 1 meter on the map G caused the emergence of alternative graph placements from 1 to 10. Accuracy up to 5 meters on given map causes stable appearance of more than 40 alternative graph placements over the entire range of the simulation parameters.

**5.7 Results**

The results of simulations showed performance of the proposed algorithm, but also the fact that the measurement errors and the lack of line of sight between the nodes in the network may lead to non-unique alternative placements or uncertain results of localization by embedding the distances graph on a digital map. In addition, you will notice that there is a certain critical density of the network, above which the quality of the localization increases significantly and remains stable.

Intuitively it is expected and is clear that the localization accuracy should increase with the density of the network increasing. Increasing the number of neighbor nodes for each node means more constraints to embed the distance graph of by spatial query. However, after a certain critical density further improvement does not occurs any more. Moreover, given the considerable measurement error, the network with high density will be more susceptible to alternative placements.

It should be noted that almost all of the various localization algorithms proposed for wireless networks have much in common in their structure:

- Measure the distance between the non-localized nodes and anchors;
- Identify the location of the node relative to the anchor;
- Optionally, accurate coordinates of the nodes using the location data of neighboring nodes.

The first phase is most often implemented by algorithms such as Sum-dist, DV-hop, or the Euclidean distance propagation. However, all of these algorithms for the successful operation require at least three anchors and rigid graph of distances between nodes.

That is the reason why the proposed method of scene analysis using spatial queries deserves special attention as a method which allows localizing objects in cases where other methods are not applicable for fundamental reasons. Of particular interest in this regard are scene analysis techniques, combined with the placement of the graph distances and terrain maps. This feature allows the use of standard digital maps, which are usually equipped with transport navigation systems and existing communication systems.

However, the proposed realization of the described localization method on the basis of spatial queries can solve the problem of high computational complexity, what is typical for scene analysis methods. All this makes it possible to translate the problem of the transportation systems mobile objects localization in terms of limited spatial data availability from theoretical considerations to practical implementation.

This chapter has shown a possible implementation of the localization algorithm of wireless networks with additional information resources. Described in detail features of the implementation and customization used for the practical problem solution, based on spatial queries on Oracle Spatial 11g platform. The construction of the simulation model of different scenarios localization has been shown. Specific scenarios and parameters for the study and analysis of the algorithm in this implementation have been selected. Conducted a series of experiments, collected and processed data, what allowed characterizing more accurately the range of problems and opportunities for application of the proposed algorithm. Considered and limitations and constraints in its application.

# CONCLUSION

This thesis is devoted to problems of objects localization in corporate transportation networks. Particular attention is paid to the methods of localization of ad-hoc networks with additional alternative sources of information in terms of inability to use global positioning systems.

Relevance of the work is determined by the need of mobile objects reliable positioning in a variety of environmental conditions. Unfortunately, most of the “classical” methods to solve this problem do not achieve the desired result in all possible embodiments and traffic environment. Some new perspectives in solving this problem reveals the rapid development and spread of autonomous intelligent devices and communications equipment, thanks to which there are additional opportunities for localization of objects equipped with such devices, by cooperative interaction of such objects.

At present, it is known a significant amount of research devoted to the new heuristic localization methods beyond the scope of classical computational geometry. Such methods are known as methods of the scene analysis. Nevertheless, many fundamental localization problems based on the analysis of the topology of the scene remains unsolved.

During the research following results were achieved:

1. The review of the classical methods of localization, the most frequently used in transport systems. Review and classification provided for the most promising, from the point of view of the author, new approaches to solving the problems of localization. It includes also a description of advantages, disadvantages and features of these approaches and their implementation methods;
2. The detailed analysis of localization system requirements in ad-hoc corporate transportation networks. The features associated with the various conditions of member access to the network information resources and the dynamic changes in the network configuration. Defined types of relevant software applications and their sensitivity to coordinates accuracy and actuality;
3. Evidences were provided that in the corporate transportation systems, depending on the type of application, both the relative and the absolute global coordinates of nodes may be relevant. Also, it is noted that the localization method should consider that the network within a long period time, may stay both static and dynamically changing. Furthermore, due to network reconfiguration serious constraints can be imposed on the possibilities of distributed computing. However, the most serious limiting factor in the application of classical localization methods is generally a high probability of insufficient number of reference points or lack of precise topological data required for localization. At the same time, devices with online access to the

information network opens up new alternatives for obtaining missing data;

4. Justified the choice of the localization method, based on the scene analysis, by enabling the collection and use of data on the possible location and network location context to reconstruct temporarily unavailable or create new reference points, as well as receive other additional geometric data;
5. Formulated and developed a model based on spatial database, what provides support for digital maps prototyping. Support includes the ability to store topological, network and other specific objects metadata, the ability for debugging and profiling. This model provides a good opportunity to analyze different localization scenarios of transportation networks;
6. Developed the method of localization by placing a graph given by the matrix of distances between localizable objects on a graph, describing the environment to which localizable objects belongs, when all the vertices of the distance graph are located on the edges or match with the vertices of the environment;
7. The method is able to use digital maps and other additional data available in GIS systems, in order to create constraints that increase rigidity of graph being embedded and thereby eliminate uncertainty and excluding alternative placements. In some scenarios, the method may preserver its operability even when there is only one reference point whose coordinates are known;
8. Conducted validation and analysis of the results of applying the developed method for different localization scenarios. The above scenarios included networks of different size, the most common configuration fragments of digital maps, emulation of lack of line of sight between network nodes and various distance measurement errors.

The experimental results suggest that the topological data provided by modern GIS-systems can be effective alternative information resource for decision-making support in cases when of the classical methods of objects localization in the cooperative transport systems are not applicable. Positioning based on a combination of classical methods and spatial queries, helps to compensate the possibility of temporary dysfunction of GPS, as well as, if necessary, correct the errors encountered in the other localization methods.

The results of the thesis can serve as the basis for further improvement of scene analysis methods. The basis for this improvement is use of not only topological but other additional information resources as embedding constraints. The most promising in this respect are the following resources:

- Exogenous resources, where each localization scene has its patterns of signal propagation. Information on such patterns stored in the dedicated spatial database layer can be transparently included in the topological queries;

- Patterns of objects that take into account the type of vehicle and its inherent size and other parameters. Data transmission on the type of vehicle within the localization application communication protocol may be used for further measurement calibration and error compensation;
- Information about the objects proximity. Proximity factor, without any significant improvements can be estimated by spatial query and can serve as an additional indicator that eliminates some alternative placements if they arise as a result of the proposed localization method;
- Information about the line of sight between the network nodes. The experiments showed that the proposed method, in some cases, might demonstrate good results with a very high proportion (up to 90%) of non-line-of-sight between network nodes. At the same time, the lack of direct line of sight between the network nodes, with high probability occurs due to the presence of obstacles. If the digital map data is not only about the road infrastructure, but also about objects that can serve as obstacles, the data that are associated with data visibility, could be another, an additional set of constraints with lower priority. With this improvement, it becomes possible to locate the network with weakly connected distance graphs where other localization methods usage is very difficult or impossible in principle;
- Infrastructure resources. These include extensive computational capabilities of modern transportation infrastructure IT hardware, allowing to identify store and share data on the vehicles involved in the localization and, further, to take into account their movement between scenes.

Mentioned above ways of improving the localization software solutions will bring the quality of applications solving reliability, safety and efficiency problems of cooperative systems in transportations telematics to higher level and will simplify its implementation.

## BIBLIOGRAPHY


[1]. Ambrose, J.K., Bukovsky, D.J., Sedlak, T.J. and Goeden, S.J. *Developing a Travel Route Planner Accounting for Traffic Variability*. Proceedings of Systems and Information Engineering Design Symposium SIEDS'09, Charlottesville, 24-24 April 2009, pp. 264–268.

[2]. Tao, H.Q., Liu, W.C. and Ma, S.B. *Intelligent Transportation Systems for Wireless Sensor Networks Based on ZigBee.* Proceedings of International Conference on Computer and Communication Technologies in Agriculture Engineering (CCTAE2010), 2010, pp. 396–399.

[3]. Monahan, Torin. *"War Rooms" of the Street: Surveillance Practices in Transportation Control Centers.* The Communication Review 10 (4): 2007, pp. 367–389.

[4]. Abdalla G.M.T and Abu - Rgheff. M.A. *Current trends in vehicular ad hoc networks.* In Proc. IEEE Global Inf. Infrastruct. Symp. Marrakech, Morocco, 2007, pp. 1–9.

[5]. Research and Innovative Technology Administration. *ITS Strategic Research Plan, 2010-2014 Fact Sheet*.
Accessed from: http://www.its.dot.gov/strat_plan/pdf/2010_factsheet.pdf [March 2, 2010].

[6]. Research and Innovative Technology Administration. *Intelligent Transportation Systems – IntelliDrive*. Accessed from: http://www.its.dot.gov/intellidrive/index.htm [March 3, 2010].

[7]. Hwang, M., J. Kemp, E. Lerner-Lam, N. Neuerburg, and P. Okunieff. *Advanced Public Transportation Systems: State of the Art Update 2006*. FTA, U.S. Department of Transportation, Washington, D.C. 2006.

[8]. Caragliu, A., Del Bo, C. and Nijkamp, P. *Smart Cities in Europe*. Proceedings of the 3rd Central European Conference in Regional Science – CERS 2009, Košice, 7-9 October 2009, pp. 49–59.

[9]. Al-Sultan, S., Al-Bayatti, A.H. and Zedan, H. *Context-Aware Driver Behavior Detection System in Intelligent Transportation Systems*. IEEE Transactions on Vehicular Technology, 62, 2013, pp. 4264–4275.

[10]. Katiyar, V., Kumar, P. and Chand, N. *An Intelligent Transportation Systems Architecture Using Wireless Sensor Networks. International Journal of Computer Applications*, 14, 2011, pp. 22–26.

[11]. R. Michoud, A. M. Orozco, and G. Llano. *Mobile ad-hoc routing protocols survey for the design of VANET applications.* Intelligent Transportation Systems Symposium (CITSS), 2012 IEEE Colombian, 2012, pp. 1–6.

[12]. P. Muhlethaler, Y. Toor, A. Laouiti, and A. de La Fortelle. *Vehicle ad hoc networks: applications and related technical issues.* IEEE Communications Surveys and Tutorials, vol. 10, no. 3, 2008, pp. 74–88.

[13]. P. Papadimitratos, A. de La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza. *Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation.* Communications Magazine, IEEE, vol. 47, no. 11, November 2009, pp. 84–95.

[14]. ISO 21217:2014 *Intelligent transport systems – Communications access for land mobiles (CALM) – Architecture*. March 2014.

[15]. ISO/DTS 17423 *Intelligent transport systems – Cooperative systems - ITS application requirements for automatic selection of communication interfaces.* February 2013.

[16]. ISO/DTS 17419 *Intelligent transport systems – Cooperative systems - Classification and management of ITS applications in a global context*. February 2013.

[17]. ISO/NP 17429 *Intelligent transport systems – Cooperative systems – Profiles for processing and transfer of information between ITS stations for applications related to transport infrastructure management, control and guidance*. December 2012.

[18]. S. Biswas, R. Tatchikou, F. Dion. *Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety*. IEEE Communications Magazine 44 (1), 2006, pp. 74–82.

[19]. J. Blum, A. Eskandarian, L. Hoffman. *Challenges of inter-vehicle ad hoc networks, IEEE Transactions on Intelligent Transportation Systems* 5 (4), 2004, pp. 347–351.

[20]. W. Kiess, J. Rybicki, M. Mauve. *On the nature of inter-vehicle communication*. In: WMAN 2007: Proceedings of the 4th Workshop on Mobile Ad-Hoc Networks, 2007, pp. 493–502.

[21]. Ferris, B., Watkins, K., and Borning, A. *Location-Aware Tools for Improving Public Transit Usability.* IEEE Pervasive Computing. Vol. 9, no. 1, Jan-Mar 2010, pp. 13–19.

[22]. R.Parker and S.Valaee. *Vehicle Localization in Vehicular Networks*. Vehicular Technology Conference, 2006. VTC-2006 fall. IEEE 64th, pp. 1–5.

[23]. A.Boukerche, H.A.B.F.Oliveira and E.F.Nakamura. *Vehicular Ad Hoc Networks: A New Challenge for Localization-Based Systems*. Computer Communications, Vol. 31, No. 12. 30 July 2008, pp. 2838–2849.

[24]. A. Thangavelu, K. Bhuvaneswari, K. Kumar, K. SenthilKumar, S.Sivanandam. *Location identification and vehicle tracking using vanet (vetrac).* In: International Conference on Signal Processing, Communications and Networking, 2007. ICSCN’07, 2007, pp. 112–116.

[25]. J. Levinson and S. Thrun. *Robust vehicle localization in urban environments using probabilistic maps.* In Proc. IEEE Int. Conf. Robot. and Automation, Anchorage, AK, May 2010, pp. 4372–4378.

[26]. T. Wu and A. Ranganathan. *Vehicle localization using road markings*. In IEEE Intell. Vehicles Symp., June 2013, pp. 1185–1190.

[27]. A. Ranganathan, D. Ilstrup, and T. Wu. *Light-Weight localization for vehicles using road markings.* In Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst., Tokyo, Japan, Nov. 2013, pp. 921–927.

[28]. A. Stewart and P. Newman. *LAPS – localisation using appearance of prior structure: 6-dof monocular camera localisation using prior pointclouds.* In Proc. IEEE Int. Conf. Robot. and Automation, Saint Paul, MN, May 2012, pp. 2625–2632.

[29]. T. Senlet and A. Elgammal. *A framework for global vehicle localization using stereo images and satellite and road maps.* In Proc. IEEE Int. Conf. Comput. Vis. Workshops, Barcelona, Spain, Nov. 2011, pp. 2034–2041.

[30]. A. Kitanov, S. Bisevac, and I. Petrovic. *Mobile robot self-localization in complex indoor environments using monocular vision and 3D model*. In IEEE/ASME Int. Conf. Advanced Intell. Mechatronics, Zurich, Switzerland, Sept. 2007, pp. 1–6.

[31]. J. Lindenberg, W. Pasman, K. Kranenborg, J. Stegeman and M. Neerincx. *Improving service matching and selection in ubiquitous computing environments: a user study.* Personal and Ubiquitous Computing. Springer London, vol 11, 2007, pp. 59–68.

[32]. Dru, M-A. and Saada, S. *Location-based mobile services: The essentials.* Alcatel Telecommunications Review, first quarter, 2001, pp. 71–76.

[33]. Geer D. *The E911 dilemma.* Wireless Business and Technology, Nov./Dec. 2001, pp. 40–44.

[34]. Zhang Yun, Gao Ren, Bian Fuling. *A Conceptual Architecture for Advanced Location Based Services in 4G Networks.* IEEE Wireless Communications, Networking and Mobile Computing, Shanghai, China, 2007, pp. 6525–6528.

[35]. Arpan Roy, Adway Mitra, Haimasree Bhattacharya, Shibasis Biswas and P.K. Das. *A Distributed Approach to Geographical Information System (GIS) and Location based Service for Internet-enabled handheld Mobile Device*s. International Conference on Wireless, Mobile and Multimedia Networks, Mumbai, India, 2008, pp. 192–195.

[36]. Zanuldin Ahmad, Sellappan Palaniappan, Woon Wei Lee, Yew Kok Han. *Geographical Information System-supported location based services: Seamless content & functionality integration using web services*. International Conference on Computing and Informatics, Kuala Lumpur, Malaysia, 2006, pp. 1–6.

[37]. José R, Moreira A, Rodrigues H and Davies N. *The AROUND architecture for dynamic location-based services*. Mobile Networks and Applications 8(4), 2003, pp. 377 – 387.

[38]. Li C and Longley P. *A test environment for Location Based Services Applications*. Transactions in GIS 10 (1), 2006, pp. 43–61.

[39]. Wealands K, Miller S, Benda P, Cartwright W. *User Assessment as Input for Useful Geospatial Representations within Mobile Location-Based Services*. Transactions in GIS 11(2), 2007, pp. 283–309.

[40]. Z. Alexander and J. Matthias. *Implementing adaptive mobile GI services based on ontologies: Examples from pedestrian navigation support.* Computers, Environment and Urban Systems 30 (6), 2006, pp. 784–798.

[41]. *Roadmap to a Single European Transport Area – Towards a competitive and resource efficient transport system*. European Institutions, Member States and stakeholders, White paper, 2011, pp. 1–17.

[42]. Nabil Drawil, Otman Basir. *Intervehicle-Communication-Assisted Localization.* IEEE Transactions on Intelligent Transportation Systems, vol.11, no.3, Sept. 2010, pp. 678–691.

[43]. Sliety M. *Impact of Vehicle Platform on Global Positioning System Performance in Intelligent Transportation*. Intelligent Transport Systems, IET 1(4), 2007, pp. 241–248.

[44]. Stockus, A., Bouju, A., Bertrand, F. & Boursier, P. *Web-Based Vehicle Localization*. Proceedings of the IEEE Intelligent Vehicles Symposium, 2000, pp. 436–441.

[45]. Skog, I. & Handel, P. *In-Car Positioning and Navigation Technologies–A Survey*. IEEE Transactions on Intelligent Transportation Systems 10(1), 2009, pp. 4–21

[46]. Fuerstenberg, K. & Weiss, T. *Feature-Level Map Building and Object Recognition for Intersection Safety Applications*. Proceedings of IEEE Intelligent Vehicles Symposium, 2005, pp. 490–495.

[47]. Jabbour M, Bonnifait P, Cherfaoui V. *Enhanced Local Maps in a GIS for a Precise Localization in Urban Areas.* Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE, 17-20 Sept. 2006, pp. 468-473.

[48]. Rae A. & Basir O. *A Framework for Visual Position Estimation for Motor Vehicles*. 4th Workshop on Positioning, Navigation and Communication, 2007, pp. 223–228.

[49]. Weiss T., Kaempchen N. & Dietmayer K. *Precise ego-Localization in Urban Areas using Laserscanner and High Accuracy Feature Maps*. Proceedings of IEEE Intelligent Vehicles Symposium, 2005, pp. 284–289.

[50]. P. Bahl and V. Padmanabhan. *RADAR: An In-Building RF-Based User Location and Tracking System.* Proc. IEEE Infocom 2000, IEEE CS Press, Los Alamitos, Calif., 2000, pp. 775–784.

[51]. F. Forno, G. Malnati, and G. Portelli. *Design and implementation of a bluetooth ad hoc network for indoor positioning.* IEE Proceedings: Software, v 152, n 5, October 2005, Microsoft Research Embedded Systems, 2007. pp 223–228.

[52]. A. Hevner, S. Chatterjee. *Design Research in Information Systems*. Integrated Series in Information Systems 22, Springer Science+Business Media LLC, 2010, pp. 14–22.

[53]. F. Ducatelle and L. M. Gambardella G. D. Caro. *A study on the use of manets in an urban environment.* IDSIA / USI-SUPSI. Technical Report No. IDSIA-01-07, 2007, pp. 1 – 12.

[54]. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci. *Wireless sensor networks: a survey*. Computer Networks 38, 2002, pp. 393–422.

[55]. R. O. LaMaire. *Wireless LANs and Mobile Networking: Standards and Future Directions*. IEEE Commun. Mag., vol. 34, no. 8, 1996, pp. 86–94.

[56]. K. C. Chen. *Medium Access Control of Wireless LANs for Mobile Computing.* IEEE Network, vol. 8, no. 5, 1994, pp. 50–63.

[57]. Part 11: *Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) specifications*: Amendment 7: *Medium Access Control (MAC) Quality of Service (QoS) Enhancements*. IEEE Draft, Rev.D12.0, 2004.

[58]. J. G. Andrews, Ph. D and R. Mohamed. *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*. Upper Saddle River, NJ: Prentice Hall, 2007, pp. 33–61.

[59]. Mishra, Ajay K. *Fundamentals of Cellular Network Planning and Optimization, 2G/2.5G/3G…Evolution of 4G*. John Wiley and Sons, 2004, pp. 3–6.

[60]. Toh C. K. *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice Hall, New Jersey, USA, 2002, 574–581.

[61]. Heikki Karjaluoto. *An Investigation of Third Generation (3G) Mobile Technologies and Services.* University of Oulu, Contemporary Management Research, Vol.2, No.2, September 2006, pp. 91–104.

[62]. R. Prasad, T. Ojanpera. *An Overview of CDMA Evolution toward Wideband CDMA.* IEEE communications Surveys, vol. 1, no. 1, Fourth Quarter 1988, pp. 2–28.

[63]. Directive 2010/40/EU of the European Parliament and of the Council of 7 July 2010, pp. 207/9–207/13.

[64]. Technical Description and Identification of Policy and Institutional Issues. *An Approach to Communications Security for a Communications Data Delivery System for V2V/V2I Safety.* www.its.dot.gov/index.htm, White Paper, November 2011.

[65]. J. Kenney. *Dedicated Short Range Communication Applications Tutorial, IEEE 802.11-3/0541r1*. 14 May 2013. Accessed from: https://mentor.ieee.org/802.11/dcn/13/11-13-0541-01-0wng-dsrc-applicationstutorial.pptx [August 29, 2013].

[66]. J. Lansford and J. Kenney. *Coexistence issues between 802.11p and 802.11ac in the proposed UNII-4 band, IEEE 802.11-13/0552r0.* 14 June 2013. Accessed from: https://mentor.ieee.org/802.11/dcn/13/11-13-0552-00-0wng-802-11pdsrc-and-802-11ac-coexistence.ppt [September 13, 2013].

[67]. Wassim G. Najm, Jonathan Koopmann, John D. Smith, and John Brewer. *Frequency of Target Crashes for IntelliDrive Safety Systems.* U.S. Department of Transportation Research and Innovative Technology Administration John A. Volpe National Transportation Systems Center Cambridge, MA 02142. October 2010, pp. 5–24.

[68]. Xin Wang. *Mobile Ad-Hoc Networks: Applications.* InTech, Chapters published January 30, 2011, pp 125–148.

[69]. Wesley M. Eddy. *At What Layer Does Mobility Belongs?,* IEEE Communication Magazine, Oct 2004, pp. 155–159.

[70]. W. Holfelder. *Vehicle-to-Vehicle and Vehicle-to-Infrastructure Communication Recent Developments, Opportunities and Challenges*. DaimlerChrysler Research and Technology North America, Inc. Palo Alto, CA. Future Generation Software Architectures in the Automotive Domain Connected Services in Mobile Networks La Jolla, CA January 10-12, 2004, pp. 3–26.

[71]. Yunhao Liu, Zheng Yang. Location, *Localization, and Localizability Location-awareness Technology for Wireless Networks*. Springer New York Dordrecht Heidelberg London, 2010, pp 112–114.

[72]. David Munoz, Frantz Bouchereau Lara, Cesar Vargas, Rogerio Enriquez-Caldera. Position Location Techniques and Applications. Academic Press, 2009, pp. 62–107.

[73]. Guoqiang Mao, Baris Fidan. *Localization Algorithms and Strategies for Wireless Sensor Networks.* Information Science Reference, Book News, Inc., Portland. 2009. pp. 57–73.

[74]. Tolga Eren, Walter Whiteley, Peter N. Belhumeur. *A theoretical analysis of the conditions for unambiguous node localization in sensor networks*. Department of Computer Science, Columbia University, 2004. Accessed from: http://www.cs.columbia.edu/techreports/cucs-032-04.pdf [September 14, 2013]

[75]. Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, Mani B. Srivastava. *Coverage Problems in Wireless Ad-hoc Sensor Networks.* IEEE INFOCOM 2001, pp. 1380–1387.

[76]. Azwirman Gusrialdi, Sandra Hirche, Takeshi Hatanaka, Masayuki Fujita. *Voronoi based coverage control with anisotropic sensors*. American Control Conference, 11-13 June 2008, pp. 736–741.

[77]. Azwirman Gusrialdi, Lu Zeng. *Distributed Deployment Algorithms for Robotic Visual Sensor Networks in Non-Convex Environment.* Networking, Sensing and Control (ICNSC), IEEE International Conference on 11-13 April 2011, pp 445–450.

[78]. Rainer Baumann. *Vehicular Ad hoc Networks (VANET) (Engineering and simulation of mobile ad hoc routing protocols for VANET on highways and in cities)*. Master's Thesis in Computer Science, ETH Zurich 2004, pp. 7–11, 76.

[79]. Nabil Drawil and Otman Basir. *Emerging New Trends in Hybrid Vehicle Localization Systems, Global Navigation Satellite Systems: Signal, Theory and Applications.* Shuanggen Jin (Ed.), InTech: 2012, pp 279–298.

[80]. Jeffrey Hightower and Gaetano Borriello. *Survey and Taxonomy of Location Systems for Ubiquitous Computing*. University of Washington, Computer Science and Engineering Box 352350, Seattle, WA 98195 Technical Report UW-CSE 01-08-03, August 24, 2001, pp. 1–29.

[81]. Jeffrey Hightower, Gaetano Borriello. *Location Systems for Ubiquitous Computing*. IEEE Computer Society Press Los Alamitos, CA, USA, Journal Computer archive Volume 34 Issue 8, August 2001, pp. 57–66.

[82]. B. Brumitt. *Ubiquitous Computing and the Role of Geometry*. IEEE Personal Comm., Oct. 2000, pp. 41–43.

[83]. J. Barton and T. Kindberg. *The CoolTown User Experience*. Tech. report 2001-22, HP Laboratories, Palo Alto, Calif., 2001.

[84]. Roy Want, Andy Hopper, Veronica Falcão, Jonathan Gibbons. *The active badge location system.* ACM Transactions on Information Systems (TOIS), v.10 n.1, Jan. 1992, pp. 91–102.

[85]. A. Harter and A. Hopper. A Distributed Location System for the Active Office. IEEE Network, Jan./Feb. 1994, pp. 62–70.

[86]. Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, Paul Webster. *The anatomy of a context-aware application*. Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, Seattle, Washington, USA, August 15-19, 1999, pp. 59–68.

[87]. Nissanka B. Priyantha, Anit Chakraborty, Hari Balakrishnan. *The Cricket location-support system.* Proceedings of the 6th annual international conference on Mobile computing and networking, Boston, Massachusetts, USA , August 06-11, 2000, pp. 32–43.

[88]. F. Raab. *Magnetic Position and Orientation Tracking System*. IEEE Trans. Aerospace and Electronic Systems, Sept. 1979, pp. 709–717.

[89]. *Technical Description of DC Magnetic Trackers.* Ascension Technology Corp. Burlington Vt. 2001.

[90]. S.R. Bible, M. Zyda, and D. Brutzman. *Using Spread-Spectrum Ranging Techniques for Position Tracking in a Virtual Environment.* Second IEEE Workshop Networked Realities. Accessed from: http://www.npsnet.org/~zyda/pubs/NR95-Paper-Bible.pdf [September 13, 2013].

[91]. John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, Steve Shafer. *Multi-Camera Multi-Person Tracking for EasyLiving*. Proceedings of the Third IEEE International Workshop on Visual Surveillance (VS'2000), July 01-01, 2000, p. 3.

[92]. T. Darrell, G. Gordon, M. Harville, J. Woodfill. *Integrated Person Tracking Using Stereo, Color, and Pattern Detection*. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 23-25, 1998, p. 601.

[93]. Robert J. Orr, Gregory D. Abowd. *The smart floor: a mechanism for natural user identification and tracking*. CHI '00 Extended Abstracts on Human Factors in Computing Systems, The Hague, The Netherlands, April 01-06, 2000, pp. 275–276.

[94]. Dieter Fox, Wolfram Burgard, Hannes Kruppa, Sebastian Thrun. *A Probabilistic Approach to Collaborative Multi-Robot Localization*. Autonomous Robots, v.8 n.3, June 2000, pp. 325–344.

[95]. L. Doherty. Convex Position Estimation in Wireless Sensor Networks. Proc. Infocom 2001, IEEE CS Press, Los Alamitos, Calif., 2001, pp. 1655–1663.

[96]. N. Bulusu J. Heidemann and D. Estrin. *GPS-Less Low-Cost Outdoor Localization for Very Small Devices*. Special Issue on Smart Spaces and Environments, IEEE Personal Comm., Oct. 2000, pp. 28–34.

[97]. J. Hightower R. Want and G. Borriello. *SpotON: An Indoor 3d Location Sensing Technology Based on RF Signal Strength*. UW CSE 2000-02-02, Univ. Washington, Seattle, Feb. 2000, pp. 1–15.

[98]. Markus Bylund, Fredrik Espinoza. *Using Quake III Arena to simulate sensors and actuators when evaluating and testing mobile services*. CHI '01 Extended Abstracts on Human Factors in Computing Systems, March 31-April 05, 2001, Seattle, Washington, 2001, pp. 241–242.

[99]. Krebss V., Tsilker B. *VANET object localization specifics*. In: Proceedings of the 13th International Conference “Reliability and Statistics in Transportation and Communication” (RelStat'13), 16–19 October 2013, Riga, Latvia, p. 249–256.

[100]. Lance Doherty, Kristofer S. J. Pister, and Laurent El Ghaoui. *Convex Position Estimation in Wireless Sensor Networks.* In INFOCOM '01: Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies, volume 3, 2001, pp. 1655–1663.

[101]. Feng Zhao, Leonidas Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann, 2004, pp. 7–8.

[102]. Department of defense. *Global Positioning System, Standard Positioning System Service, Signal Specification.* 2nd Edition, 1995, p. 18.

[103]. Corn, David. *Surveying By Satellite*. Catalyst, February 1990, pp. 43–44.

[104]. Denaro R. P. and R. Kalafus. *Advances and Test Results in Differential GPS Navigation*. The Journal of Navigation, vol. 43:1, 1990, pp. 32–40.

[105]. McDonald, K. D. *GPS Progress and Issues*. GPS World, Vol. 1:1, 1990, p. 16.

[106]. Trimble Navigation Limited. *A New Level of Accuracy for Differential GPS Mapping Applications Using EVERESTJ Multipath Rejection Technology*. Mapping and GIS Systems Division. Document #102, 1997, pp. 1–7.

[107]. A. Savvides, H. Park, M. Srivastava. *The bits and flops of the N-hop multilateration primitive for node localization problems*. First ACM International Workshop on Wireless Sensor Networks and Application, Atlanta, GA, 2002. pp. 112–121.

[108]. C. Saad, A. Benslimane, J. Konig. *AT-Dist: A Distributed Method for Localization with High Accuracy in Sensor Networks*. Computers and Communications, ISCC 2008. IEEE Symposium on 6-9 July 2008, pp 1190–1195.

[109]. D. Niculescu and B. Nath. *Ad-hoc positioning system*. In Global Telecommunications Conference, GLOBECOM '01. IEEE (Volume: 5), November 2001, pp. 2926–2931.

[110]. C. Savarese, K. Langendoen, and J. Rabaey. *Robust positioning algorithms for distributed ad-hoc wireless sensor networks*. In USENIX technical annual conference, Monterey, CA, June 2002. pp 317–328.

[111]. N. Reijers, K Langendoen. *Distributed Localization Algorithms. Embedded Systems*. Handbook Edited by Richard Zurawski, CRC Press 2005. pp. 36.1–36.23.

[112]. E. Trevisani and A. Vitaletti. *Cell-id location technique, limits and benefits: an experimental study*. In Mobile Computing Systems and Applications, 2004. WMCSA 2004. Sixth IEEE Workshop on, 2004, pp. 51–60.

[113]. D. Niculescu and B. Nath. *Ad hoc positioning system (APS) using AOA*. In IEEE INFOCOM, volume 22, March 2003, pp. 1734–1743.

[114]. D. Niculescu and B. Nath. *Error characteristics of ad hoc positioning systems (APS).* In MobiHoc ’04: Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2004, pp. 20–30.

[115]. A. Savvides, C. Han, and M. B. Strivastava. *Dynamic fine-grained localization in ad-hoc networks of sensors.* In Proc. 7th Annual International Conference on Mobile Computing and Networking (MobiCom 2001), Rome, Italy, ACM Press, July 2001, pp. 166–179.

[116]. David Moore, John Leonard, Daniela Rus, and Seth Teller. *Robust distributed network localization with noisy range measurements.* In Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys'04), Baltimore, MD, November 2004, pp. 50–61.

[117]. R. Parker and S. Valaee. *A Robust Min-Max Localization*. Intelligent Transportation Systems Conference, ITSC '06. IEEE, 2006, pp. 1000–1005.

[118]. Alan Kaminsky. *Trilateration*. Rochester Institute of Technology, March 8, 2007, pp. 1–4. Accessed from: http://ahvaz.ist.unomaha.edu/azad/temp/sal/07-kaminsky-lateration-trilateration-coordinate-localization-sensors-good.pdf [September 14, 2013].

[119]. William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press New York, NY, USA, 1992.

[120]. M. J. Feuerstein. Urban and Indoor Location using Pattern Matching of Wireless Network Measurements. Invited Workshop on Opportunistic RF Localization for Next Generation Wireless Devices, Worcester Polytechnic Institute, June 2008. pp. 1–5.

[121]. Jing Wang, R. K. Ghosh, Sajal K. Das. *A survey on sensor localization.* Journal of Control Theory and Applications, January 2010, pp. 8.1–2.11.

[122]. Nissanka B. Priyantha, Anit Chakraborty, Hari Balakrishnan. *The Cricket Location-Support System.* In MobiCom '00: Proceedings of the 6th ACM/IEEE International Conference on Mobile Computing and Networking, ACM Press New York, NY, USA, 2000. pp. 32–43.

[123]. Joon-Yong Lee, Robert A. Scholtz. *Ranging in a Dense Multipath Environment Using an UWB Radio Link.* IEEE Journal on Selected Areas in Communications, 20(9), 2002, pp. 1677–1683.

[124]. Jun-geun Park B.S., Seoul National University. *Moving-Baseline Localization for Mobile Wireless Sensor Networks*. Massachusetts institute of technology, February 2009, pp. 33–55.

[125]. Mathieu Bouet, Aldri L. dos Santos. *Wireless Days.* WD '08. 1st IFIP, 24-27 November 2008, pp. 1–5.

[126]. Eric Richter, Marcus Obst, Veit Leonhardt, Robin Schubert, Norman Mattern, Gerd Wanielik. *Cooperative Localization Algorithms for Improved Road Navigation*. Royal Institute of Navigation Annual Conference and Exhibition 2008 and 37th ILA Annual

Convention and Technical Symposium (NAV 08/ILA 37) London, UK 28-30, October 2008, pp. 506–511.

[127]. K. Madsen, H.B. Nielsen, O. Tingleff. *Methods for non-linear least squares problems*. Informatics and Mathematical Modelling Technical University of Denmark. 2nd Edition, April 2004, pp 24–29.

[128]. Savides A., Garber W., Adlakha S., Moses R., Srivastava B. *On the Error Characteristics of Multihop Node Localization in Ad-Hoc Sensor Networks*. In: IPSN'03 Proceedings of the 2nd international conference on Information processing in sensor networks. Springer, 2003, pp 317–332.

[129]. Juan Liu, Ying Zhang, and Feng Zhao. *Robust distributed node localization with error management*. In: MobiHoc '06: Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing, New York, ACM Press, 2006, pp. 250–261.

[130]. Zheng Yang, Yunhao Liu. *Quality of Trilateration: Confidence based Iterative Localization*. In: IEEE Transactions on Parallel and Distributed Systems, Vol. 21, No 5, May 2010, pp. 631–640.

[131]. Dragos Niculescu, Badri Nath. *Position and orientation in ad hoc networks.* Department of Computer Science, Rutgers University, Piscataway, NJ 08854, USA. 20 July 2003, pp 6–12.

[132]. James Newsome, Dawn Song. *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.

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

[134]. Nissanka B. Priyantha, Hari Balakrishnan, Erik Demaine, Seth Teller. *Anchor Free Distributed Localization in Sensor Networks.* Tech Report #892, April 15, 2003 MIT Laboratory for Computer Science. Accessed from: http://nms.lcs.mit.edu/cricket/ [April 8, 2013].

[135]. B. Hendrickson. *Conditions for unique graph realizations*. SIAM Journal of Computing, vol. 21, no. 1, 1992, pp. 65–84.

[136]. G. Laman. *On graphs and rigidity of plane skeletal structures.* Journal of Engineering Mathematics, vol. 4, 1970, pp. 331–340.

[137]. B. Jackson, T. Jordan. *Connected rigidity matroids and unique realizations of graphs.* Journal of Combinatorial Theory Series B, vol. 94, no. 1, 2005, pp. 1–29.

[138]. J.E. Hopcroft, R.E. Tarjan. *Finding the triconnected components of a graph.* Department of Computer Science, Cornell University TR 140, 1972, pp. 3–18.

[139]. D.J. Jacobs, B. Hendrickson. *An algorithm for two-dimensional rigidity percolation: the pebble game.* Journal of Computational Physics, vol. 137, 1997, pp. 346–365.

[140]. James Aspnes, Tolga Eren, David K. Goldenberg, A. Stephen Morse, Walter Whiteley, Yang Richard Yang, Brian D. O. Anderson, Peter N. Belhumeur. *A Theory of Network Localization.* IEEE Transactions on Mobile Computing archive Volume 5 Issue 12, December 2006, pp. 1663–1678.

[141]. Technical Description. *ESRI Shapefile*. An ESRI White Paper-July 1998. Accessed from: https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf [April 9, 2013].

[142]. T. King, H. Fusler, M. Transier, W. Effelsberg. *Dead reckoning for position-based forwarding on highways.* In: Proceedings of the 3rd International Workshop on Intelligent Transportation (WIT 2006), Hamburg, Germany, 2006, pp. 199–204.

[143]. *Scalable Vector Graphics (SVG).* 1.2 W3C Working Draft, § 4.2-Drawing Order. World Wide Web Consortium, 15 November 2002. Accessed from: http://www.w3.org/TR/2002/WD-SVG12-20021115/#drawingorder [August 29, 2012].

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

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

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

[147]. Chuck Murray. *Oracle Spatial User's Guide and Reference.* 10g Release 2 (10.2) B14255-03 Oracle, 2006, pp. 29–99.

[148]. Antonin Guttman. *R-Trees: A Dynamic Index Structure for Spatial Searching*, Proc. ACM SIGMOD International Conference on Management of Data, 1984, pp. 47–57.

[149]. Arge L., De Berg, M. Haverkort, H. J., Yi, K. (2004). *The Priority R-tree.* Proceedings of the 2004 ACM SIGMOD international conference on Management of data - SIGMOD '04, 2004, p. 347.

[150]. Department of Transport, Tourism and Sport (DTTAS) and the Department of the Environment, Community and Local Government (DECLG). *Design Manual for Urban Roads and Streets.* Dublin, Ireland. March 2013, pp. 41–50.

[151]. Mahalik, Nitaigour P. *Sensor Networks and Configuration. Fundamentals, Standards, Platforms, and Applications*. Springer-Verlag Berlin Heidelberg, 2007, pp. 300–305.

# APPENDICES

## Appendix N1

The results of investigation were published in the following proceedings and journals:

1. Krebss V. Tsilker *B. Coverage Evaluation Approaches for Intelligent Transportation Systems Based On Anisotropic Sensor Networks*. Proceedings of the 10th International Conference “Reliability and Statistics in Transportation and Communication”, 2010, Riga, Latvia, pp. 199–206.
2. Krebss V. Tsilker B. *Coverage Estimation in Synergic networked Intelligent Transportation Systems for Non-Isotropic Nodes*. Transport and Telecommunication, volume 11.No.4, 2010, Riga, Latvia, pp. 36–45.
3. Krebss V., Tsilker B. *Vehicle Dynamic localization in intelligent transportation systems based on sensor networks*. Transport and Telecommunication, volume 12.No.3, 2011, Riga, Latvia, pp. 63–72.
4. Krebss V. *Analysis of the influence of the characteristics and relative position of the reference points on the error localization of objects in VANET networks*. Research and Technology – Step into the Future, volume 7. No 1, 2012, Riga, Latvia. p. 125.
5. Krebss V., Tsilker B. *VANET nodes localization with limited reference points availability*. In: Proceedings of the 12th International Conference “Reliability and Statistics in Transportation and Communication” (RelStat'12), 17–20 October 2012, Riga, Latvia, pp. 285–295.
6. Krebss V. *Spatial Query Localization Method in Limited Reference Point Environment.* International Journal of Mathematical Sciences. World Academy of Science, Engineering and Technology 75, 2013, pp 1242–1249.
7. Krebss V., Tsilker B. *VANET object localization specifics*. In: Proceedings of the 13th International Conference “Reliability and Statistics in Transportation and Communication” (RelStat'13), 16–19 October 2013, Riga, Latvia, p. 249–256.
8. Krebss V. *STSM Scientific Report improving the accuracy of real-time positioning of moving objects in mines*. COST-STSM-ECOST-STSM-IC0906-041113-036757, Luleå University of Technology, Sweden November 15, 2013.
9. Krebss V. *Spatial query localization method in limited reference point environment: an empirical study*. European Scientific Journal August 2014 edition vol.10, No.24. pp. 48–64.

# Appendix N2

Specification of key spatial functions and data structures used in the implementation of the localization method using spatial queries. Oracle® Spatial Developer's Guide 11g Release 1 (11.1) B28400-05 June 2009.

### 2.1 Function SDO_ANYINTERACT

**Format**

```
SDO_ANYINTERACT(geometry1, geometry2);
```

**Description**

Checks if any geometries in a table have the ANYINTERACT topological relationship with a specified geometry. Equivalent to specifying the SDO_RELATE operator with 'mask=ANYINTERACT'.

**Returns**

The expression SDO_ANYINTERACT(geometry1,geometry2) = 'TRUE' evaluates to TRUE for object pairs that have the ANYINTERACT topological relationship, and FALSE otherwise.

### 2.2 Function SDO_GEOM.SDO_INTERSECTION

**Format**

```
SDO_GEOM.SDO_INTERSECTION(
geom1 IN SDO_GEOMETRY,
dim1 IN SDO_DIM_ARRAY,
geom2 IN SDO_GEOMETRY,
dim2 IN SDO_DIM_ARRAY
) RETURN SDO_GEOMETRY;
or
SDO_GEOM.SDO_INTERSECTION(
geom1 IN SDO_GEOMETRY,
geom2 IN SDO_GEOMETRY,
tol IN NUMBER
) RETURN SDO_GEOMETRY;
```

**Description**

Returns a geometry object that is the topological intersection (AND operation) of two geometry objects.

**Parameters**

geom1:

Geometry object.

dim1:

Dimensional information array corresponding to geom1, usually selected from one of the xxx_SDO_GEOM_METADATA views.

geom2:

Geometry object.

dim2:

Dimensional information array corresponding to geom2, usually selected from one of the xxx_SDO_GEOM_METADATA views.

tol:

Tolerance value.

### 2.3 Function xxx_SDO_INDEX_METADATA Views

The following views contain detailed information about spatial index metadata:

- USER_SDO_INDEX_METADATA contains index information for all spatial tables owned by the user.
- ALL_SDO_INDEX_METADATA contains index information for all spatial tables on which the user has SELECT permission.

### 2.4 Function SDO_GEOMETRY Object Type

With Spatial, the geometric description of a spatial object is stored in a single row, in a single column of object type SDO_GEOMETRY in a user-defined table. Any table that has a column of type SDO_GEOMETRY must have another column, or set of columns, that defines a unique primary key for that table. Tables of this sort are sometimes referred to as spatial tables or spatial geometry tables. Oracle Spatial defines the object type SDO_GEOMETRY as:

```
CREATE TYPE sdo_geometry AS OBJECT (
SDO_GTYPE NUMBER,
SDO_SRID NUMBER,
SDO_POINT SDO_POINT_TYPE,
SDO_ELEM_INFO SDO_ELEM_INFO_ARRAY,
SDO_ORDINATES SDO_ORDINATE_ARRAY);
```

Oracle Spatial also defines the SDO_POINT_TYPE, SDO_ELEM_INFO_ARRAY, and SDO_ORDINATE_ARRAY types, which are used in the SDO_GEOMETRY type definition, as follows:

```
CREATE TYPE sdo_point_type AS OBJECT (
X NUMBER,
Y NUMBER,
Z NUMBER);
CREATE TYPE sdo_elem_info_array AS VARRAY (1048576) of NUMBER;
CREATE TYPE sdo_ordinate_array AS VARRAY (1048576) of NUMBER;
```

Because the maximum `SDO_ORDINATE_ARRAY` size is 1,048,576 numbers, the maximum number of vertices in an `SDO_GEOMETRY` object depends on the number of dimensions per vertex: 524,288 for two dimensions, 349,525 for three dimensions, and 262,144 for four dimensions. The sections that follow describe the semantics of each `SDO_GEOMETRY` attribute, and then describe some usage considerations. The `SDO_GEOMETRY` object type has methods that provide convenient access to some of the attributes.

## Appendix N3

The main classes of localization applications based on spatial queries.

### 3.1 Simulation environment initialization class

```java
import java.awt.geom.Point2D;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import java.util.Set;
import oracle.spatial.geometry.JGeometry;
import oracle.sql.STRUCT;
import org.apache.commons.collections.MultiMap;
import org.tsi.utils.GEODBUtils;

public class DBbridge {
```

```java
    private final String area = "AREA_MAP_spatial_idx";
    private final String staging = "STAGING_MAP_spatial_idx";
     // connection to the database
    final String driverName = "oracle.jdbc.driver.OracleDriver";
    final String serverName = "127.0.0.1";
    final String portNumber = "1521";
    final String sid = "orcl";
    final String url = "jdbc:oracle:thin:@" + serverName + ":" + portNumber + ":" +
sid;
    final String username = "username";
    final String password = "password";

    public DBbridge() throws ClassNotFoundException {
        super();
        Class.forName(driverName);
    }

     public void cleanDB() throws SQLException{
        //call several methods cleaning different types of objects
        this.cleanNodes();
        this.cleanMap();
     }

     public void cleanMap() throws SQLException{
        PreparedStatement pstmt = null;
        Connection connection =  getSpatialConnection();
        String query = "DELETE FROM AREA_MAP WHERE SHAPE_TYPE IN (" +
Segment.LINE.getDbid() + ", " + Segment.ARC.getDbid() + ")";
        pstmt = connection.prepareStatement(query);
        pstmt.executeUpdate();
        connection.commit();
        connection.close();
     }

     public void cleanNodes() throws SQLException{
        //segment type ID parameter (IN), bad practice, I know
        PreparedStatement pstmt = null;
        Connection connection =  getSpatialConnection();
        String query = "DELETE FROM AREA_MAP WHERE SHAPE_TYPE IN (" +
Segment.ANCHOR.getDbid() + ", " + Segment.NODE.getDbid() + ", " +
Segment.SHADOW.getDbid() + ")";
```

```java
            pstmt = connection.prepareStatement(query);
            pstmt.executeUpdate();
            connection.commit();
            connection.close();
        }

        public void insertMap(MapGeneric mg, String scheme) throws SQLException{
            PreparedStatement pstmt = null;
            Connection connection =  getSpatialConnection();
            // parameters: ID, segment name, my custom status ID,geometry, my custom
    shape type ID
            String lineQuery = "INSERT INTO AREA_MAP VALUES(NULL,:1, 1,
    SDO_GEOMETRY(2002, NULL, NULL, SDO_ELEM_INFO_ARRAY (1,2,1),
    SDO_ORDINATE_ARRAY(:2,:3,:4,:5)), 5)";
            String arcQuery = "INSERT INTO AREA_MAP VALUES(NULL, :1, 1,
    SDO_GEOMETRY(2002, null, null, sdo_elem_info_array (1,2,2), sdo_ordinate_array (:2,
    :3, :4, :5, :6, :7)), 6)";

            MultiMap multimap = mg.getMap();
            Set keySet = multimap.keySet( );
            Iterator keyIterator = keySet.iterator();

            while( keyIterator.hasNext( ) ) {
                 Segment key = (Segment)keyIterator.next( );
                Collection values = (Collection) multimap.get( key );
                Iterator valuesIterator = values.iterator( );
                int i = 1;
                while( valuesIterator.hasNext( ) ) {
                 List<Point2D> ls = (List<Point2D>)valuesIterator.next( );
                 if(key == Segment.LINE){
                      pstmt = connection.prepareStatement(lineQuery);
                      pstmt.setString(1, scheme+"segment_line_" + i);
                      pstmt.setDouble(2, ls.get(0).getX());
                      pstmt.setDouble(3, ls.get(0).getY());
                      pstmt.setDouble(4, ls.get(1).getX());
                      pstmt.setDouble(5, ls.get(1).getY());
                 }
                 else if(key == Segment.ARC){
                      pstmt = connection.prepareStatement(arcQuery);
                      pstmt.setString(1, scheme+"segment_arc_" + i);
                      pstmt.setDouble(2, ls.get(0).getX());
                      pstmt.setDouble(3, ls.get(0).getY());
```

```
                pstmt.setDouble(4, ls.get(1).getX());
                pstmt.setDouble(5, ls.get(1).getY());
                pstmt.setDouble(6, ls.get(2).getX());
                pstmt.setDouble(7, ls.get(2).getY());
            }
              pstmt.executeUpdate();
              pstmt.close();
              i++;
          }
        }
        GEODBUtils.rebuildIndex(this.area, connection);
        connection.commit();
        connection.close();
    }

    public long insertPoint(Point2D pt, String sName, Segment type) throws
SQLException{
        long pointID = -1;
        PreparedStatement pstmt = null;
        Connection connection =  getSpatialConnection();
        String query = "INSERT INTO AREA_MAP
VALUES(NULL,:1,1,SDO_GEOMETRY(2001,NULL,NULL,SDO_ELEM_INFO_ARRAY(1,1,1),SDO_ORDINATE
_ARRAY(:2,:3)), :4)";
        pstmt = connection.prepareStatement(query, new String[]{"SEGMENT_ID"});
        pstmt.setString(1, sName);
        pstmt.setDouble(2, pt.getX());
        pstmt.setDouble(3, pt.getY());
        pstmt.setDouble(4, type.getDbid());
        pstmt.executeUpdate();
        ResultSet rset = pstmt.getGeneratedKeys();
        rset.next();
        pointID = rset.getLong(1);
        connection.commit();
        connection.close();
        return pointID;
    }

    public List<Integer> getSegmenIDtList() throws SQLException{
        //returns arc and line ID's
        List<Integer> ls = new ArrayList<Integer>();
        PreparedStatement pstmt = null;
        Connection connection =  getSpatialConnection();
```

```
        String query = "SELECT SEGMENT_ID FROM AREA_MAP WHERE SHAPE_TYPE IN (" +
Segment.LINE.getDbid() + ", " + Segment.ARC.getDbid() + ")";
        pstmt = connection.prepareStatement(query);
        ResultSet rs = pstmt.executeQuery();
        while (rs.next()) {
            ls.add(rs.getInt("SEGMENT_ID"));
        }
        connection.close();
        return ls;
    }

    public List<Integer> getNodeIDtList(Segment type) throws SQLException{
        //returns arc or line ID's
        List<Integer> ls = new LinkedList<Integer>();
        PreparedStatement pstmt = null;
        Connection connection =  getSpatialConnection();
        String query = "SELECT SEGMENT_ID FROM AREA_MAP WHERE SHAPE_TYPE = :1 ORDER
BY SEGMENT_ID";
        pstmt = connection.prepareStatement(query);
        pstmt.setLong(1, type.getDbid());
        ResultSet rs = pstmt.executeQuery();
        while (rs.next()) {
            ls.add(rs.getInt("SEGMENT_ID"));
        }
        connection.close();
        return ls;
    }

   public List<Node2D> getNode2Dlist(List<Integer> ids) throws SQLException{
        List<Node2D> ls = new LinkedList<Node2D>();
        PreparedStatement pstmt = null;
        Connection connection =  getSpatialConnection();
        String query = "SELECT AREA_MAP.SEGMENT_GEO_LOCATION, AREA_MAP.SHAPE_TYPE
FROM AREA_MAP WHERE AREA_MAP.SEGMENT_ID = :1";
        for(Integer i:ids){
             pstmt = connection.prepareStatement(query);
             pstmt.setLong(1, i);
             ResultSet rs = pstmt.executeQuery();
            while (rs.next()) {
             int type = rs.getInt(2);
             Segment t = null;
             switch(type){
```

```
                    case 3:
                         t = Segment.ANCHOR;
                         break;
                    case 4:
                         t = Segment.SHADOW;
                         break;
                    case 5:
                         t = Segment.LINE;
                         break;
                         case 6:
                              t = Segment.ARC;
                         break;
                    case 7:
                         t = Segment.NODE;
                         break;
                    default:
                         throw new SQLException("Application exception. Unknown
segment type");
             }
             STRUCT st = (oracle.sql.STRUCT) rs.getObject(1);
             JGeometry j_geom = JGeometry.load(st);
             double pt[] = j_geom.getOrdinatesArray();
             double x = pt[0];
             double y = pt[1];
             Node2D nd = new Node2D(x, y, i, t);
             ls.add(nd);
            }
        }
        connection.close();
        return ls;
   }

    public double getDistance(long i, long j) throws SQLException{
        //query distance between objects with i and j id's
        double result = 0;
        PreparedStatement pstmt = null;
        Connection connection =  getSpatialConnection();
        String query = "SELECT SDO_GEOM.SDO_DISTANCE(AB.SEGMENT_GEO_LOCATION,
AC.SEGMENT_GEO_LOCATION, 0.005) AS DIST FROM AREA_MAP AB, AREA_MAP AC WHERE
AB.segment_id = :1 AND AC.segment_id = :2";
        pstmt = connection.prepareStatement(query);
        pstmt.setLong(1, i);
```

```
            pstmt.setLong(2, j);
            ResultSet rs = pstmt.executeQuery();
            while (rs.next()) {
                result = rs.getDouble("DIST");
            }
            connection.close();
            return result;
        }

        public void reindex() throws SQLException{
            Connection connection =  getSpatialConnection();
            GEODBUtils.rebuildIndex(this.area, connection);
            connection.commit();
            connection.close();
        }

        public Connection getSpatialConnection() throws SQLException{
            return DriverManager.getConnection(url, username, password);
        }
}
```

### 3.2 Measurement errors simulation

```
package org.tsi.extended;
import java.util.Random;

public class ErrorEngine {

  public double[][] distortMatrix (double in[][], double min, double max, Random
rnd){
        double result [][] = new double [in.length] [in[0].length];
        for(int i = 0; i < in.length; i++){
            for(int j = 0; j < in[i].length; j++){
                double error = min + (rnd.nextDouble() * ((max - min) + 1));
                boolean sign = rnd.nextBoolean();
                if(sign == false){
                    error = -error;
                }
                result[i][j] = in[i][j] + error;
            }
        }
        return result;
```

```
    }
}
```

### 3.3 Node placement class

```
package org.tsi.extended;
import java.awt.geom.Point2D;
import java.awt.geom.Line2D;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.LinkedList;
import java.util.List;
import java.util.Random;
import org.tsi.core.Point2DExt;
import oracle.spatial.geometry.JGeometry;
import oracle.sql.STRUCT;

public class NodeEngine {

    public final String arc = "ARC";
    public final String line = "LINE";
    private Random rnd;

    public Random getRnd() {
        return rnd;
    }

    public NodeEngine(int seed){
        rnd = new Random();
        rnd.setSeed(seed);
    }
//use this class to select (random) segments of certain kind and add to them points
at certain location, see how to combine this with BDBridge class

    public void placeRandomNodes(DBbridge dbb, int nodeCount, Segment sgType) throws
Exception{
            List<Integer> ls = dbb.getSegmenIDtList();
```

```
            Connection c = dbb.getSpatialConnection();
            Point2D result = null;
            String prefix = null;
            for(int i = 0; i < nodeCount; i++){
                 int p = rnd.nextInt(ls.size());
                 int sID = ls.get(p);
                 //getting geometry type from DB as a java object, so we can get
back original three points on arc
                 // the we can obtain arc center, radius, angle and then random
point at random angle within angle specified
                  PreparedStatement pstmt = c.prepareStatement("SELECT
AREA_MAP.SEGMENT_GEO_LOCATION, shape_types.shape_type_name FROM AREA_MAP left join
shape_types on area_map.shape_type = shape_types.shape_type_id where
SEGMENT_ID=:1");
                  pstmt.setDouble(1, sID);
                 /// reading a geometry from database
                 ResultSet rs = pstmt.executeQuery();
                 rs.next();
                 STRUCT st = (oracle.sql.STRUCT) rs.getObject(1);
                 String type = rs.getString(2);
                 //convert STRUCT into geometry
                 JGeometry j_geom = JGeometry.load(st);
                 double pt[] = j_geom.getOrdinatesArray();
                 rs.close();
                 pstmt.close();

                 if(arc.equals(type)){
                 Point2D p1 = new Point2D.Double(pt[0], pt[1]);
                 Point2D p2 = new Point2D.Double(pt[2], pt[3]);
                 Point2D p3 = new Point2D.Double(pt[4], pt[5]);

                  Point2D center = MyGeom.getArcCenter(p1, p2, p3);
                  double radius =  MyGeom.getArcRadius(p1, p2, p3);
                  Double ang [] =  {MyGeom.getAngleOfPointOnCircle(p1, center,
radius),
                  MyGeom.getAngleOfPointOnCircle(p2, center, radius),
                  MyGeom.getAngleOfPointOnCircle(p3, center, radius)};

                  double min = (double) Collections.min(Arrays.asList(ang));
                  double max = (double) Collections.max(Arrays.asList(ang));

                  double angle = min + (max -min) * rnd.nextDouble();
```

```
                    result = MyGeom.getPointAt(center, radius, angle);
                    // here we have to insert point, method present in DBbrigge, not
tested yet
                  }
                  else if(line.equals(type)){

                    Line2D l = new Line2D.Double(new Point2D.Double(pt[0], pt[1]),
new Point2D.Double(pt[2], pt[3]));
                         double r =  rnd.nextDouble();
                         double x = (1 - r)*l.getX1() + r*l.getX2();
                         double y = (1 - r)*l.getY1() + r*l.getY2();
                         result = new Point2D.Double(x, y);
                  }
                    if(sgType.getDbid() == Segment.ANCHOR.getDbid()){
                        prefix = "_Anchor";
                    }
                    else{
                         prefix = "_Free";
                    }
                  System.out.println("Inserting point at " + result.getX() + " " +
result.getY());
                  dbb.insertPoint(result, String.valueOf(i) + prefix, sgType);
             }
             if(c != null && !c.isClosed()){
                  c.close();
             }
   }

   public List<Node2D> placeShadows(DBbridge dbb, long nodeID, DMatrix dbm, double
tolerance)throws SQLException{
        //place shadows for given node, returns list of generated Node2D, TEST 1
anchor case. 2 anchor later if necessary
        Connection c = dbb.getSpatialConnection();
        List<Node2D> res = new LinkedList<Node2D>();
        Node2D base = null;
        Node2D anchor1 = null;
        Node2D anchor2 = null;
        double distance1 = 0;
        double distance2 = 0;
        /*
         find nodeID node
```

```
         */
        int num = 0;
        for(Node2D nd : dbm.getNodes()){
            if(nd.getNodeID() == nodeID){
                base = nd;
                break;
            }
            num++;
        }
        if(base == null){
            throw new SQLException("base node not found");
        }
        Point2D cpt = null;
        String query = null;
        if(dbm.getAnchors().size() == 1){ // one anchor query
        }
        else if(dbm.getAnchors().size() == 2){ // two anchor query
            throw new SQLException("NOT TESTED PROGRAMM BRANCH");
        }
        else{
            throw new SQLException("Number of anchors more than 2 or less than 1");
        }
// IMPORTANT NOTE: dynamic shape as function argument in query DOES NOT support
parameters, only string concatenation. JDBC bug?
/*
 Here find anchor Node2D and radius to nodeID node, 1 anchor case
 */
        anchor1 = dbm.getAnchors().get(0);
        // one anchor case
        distance1 = dbm.getAntrix()[0][num];

        cpt = PointOnCircle(distance1, 0, anchor1);
        double x1 = cpt.getX();
        double y1 = cpt.getY();

        cpt = PointOnCircle(distance1, 90, anchor1);
        double x2 = cpt.getX();
        double y2 = cpt.getY();

        cpt = PointOnCircle(distance1, 180, anchor1);
        double x3 = cpt.getX();
        double y3 = cpt.getY();
```

```java
        cpt = PointOnCircle(distance1, 270, anchor1);
        double x4 = cpt.getX();
        double y4 = cpt.getY();

        cpt = PointOnCircle(distance1, 360, anchor1);
        double x5 = cpt.getX();
        double y5 = cpt.getY();

        query = "SELECT Q1.SHD FROM( " +
                    "SELECT AM.SEGMENT_ID,
SDO_GEOM.SDO_INTERSECTION(AM.SEGMENT_GEO_LOCATION, SDO_GEOMETRY(2002, null, null,
sdo_elem_info_array (1,2,2), " +
                    "sdo_ordinate_array (" + x1 + ", " + y1 + ", " + x2 + ", " + y2 +
", " + x3 + ", " + y3 + ", " + x4 + ", " + y4 + ", " + x5 + ", " + y5 + ")) , :1) "
+
                    "AS SHD " +
                    "FROM AREA_MAP AM " +
                    "WHERE AM.SHAPE_TYPE IN (5,6) AND " +
                    "SDO_WITHIN_DISTANCE(AM.SEGMENT_GEO_LOCATION, (SELECT
SEGMENT_GEO_LOCATION FROM AREA_MAP WHERE SEGMENT_ID = :2), 'distance='||:3) =
'TRUE') Q1 " +
                    "WHERE Q1.SHD IS NOT NULL";// AND Q1.SHD.SDO_GTYPE = 2001";

        if(dbm.getAnchors().size() == 2){
            //two anchor case
            anchor2 = dbm.getAnchors().get(1);
            distance2 = dbm.getAntrix()[1][num];

            cpt = PointOnCircle(distance2, 0, anchor2);
            double x6 = cpt.getX();
            double y6 = cpt.getY();

            cpt = PointOnCircle(distance2, 90, anchor2);
            double x7 = cpt.getX();
            double y7 = cpt.getY();

            cpt = PointOnCircle(distance2, 180, anchor2);
            double x8 = cpt.getX();
            double y8 = cpt.getY();

            cpt = PointOnCircle(distance2, 270, anchor2);
```

```java
            double x9 = cpt.getX();
            double y9 = cpt.getY();

            cpt = PointOnCircle(distance2, 360, anchor2);
            double x10 = cpt.getX();
            double y10 = cpt.getY();

            query = "select Q1.SHD " +
"from " +
"( " +
"select AM.SEGMENT_ID, SDO_GEOM.SDO_INTERSECTION (AM.SEGMENT_GEO_LOCATION, (select
SDO_GEOM.SDO_INTERSECTION(SDO_GEOMETRY(2002, null, null, SDO_ELEM_INFO_ARRAY
(1,2,2), SDO_ORDINATE_ARRAY (" + x1 + ", " + y1 + ", " + x2 + ", " + y2 + ", " + x3
+ ", " + y3 + ", " + x4 + ", " + y4 + ", " + x5 + ", " + y5 + ")) , " +
"                     SDO_GEOMETRY(2002, null, null, SDO_ELEM_INFO_ARRAY (1,2,2),
SDO_ORDINATE_ARRAY (" + x6 + ", " + y6 + ", " + x7 + ", " + y7 + ", " + x8 + ", " +
y8 + ", " + x9 + ", " + y9 + ", " + x10 + ", " + y10 + ")), :1) AS PNT " +
"                     FROM DUAL), :1) as SHD " +
"from " +
"AREA_MAP AM " +
"WHERE AM.SHAPE_TYPE IN (5,6) AND " +
"SDO_WITHIN_DISTANCE(AM.SEGMENT_GEO_LOCATION, (SELECT SEGMENT_GEO_LOCATION FROM
AREA_MAP WHERE SEGMENT_ID = :2), 'distance='||:3) = 'TRUE' AND " +
"SDO_WITHIN_DISTANCE(AM.SEGMENT_GEO_LOCATION, (SELECT SEGMENT_GEO_LOCATION FROM
AREA_MAP WHERE SEGMENT_ID = :4), 'distance='||:5) = 'TRUE' " +
") Q1 " +
"WHERE Q1.SHD IS NOT NULL";//  AND Q1.SHD.SDO_GTYPE = 2001";
        }
        PreparedStatement pstmt = c.prepareStatement(query);
        // one anchor case
        pstmt.setDouble(1, tolerance);
        pstmt.setLong(2, anchor1.getNodeID());
        pstmt.setDouble(3, distance1);

        if(dbm.getAnchors().size() == 2){
            pstmt.setLong(4, anchor2.getNodeID());
            pstmt.setDouble(5, distance2);
        }
        int i = 0;
        ResultSet rs = pstmt.executeQuery();
        while (rs.next()) {
            STRUCT st = (oracle.sql.STRUCT) rs.getObject("SHD");
```

```
            JGeometry j_geom = JGeometry.load(st);
            double pt[] = j_geom.getOrdinatesArray();
            int k = 1; // query may give not only point but lines also, so we take
two ordinates. (polylines will work too)
            while (pt.length > k){
             double x = pt[k-1];
                  double y = pt[k];
                  String desc = "_Shadow";
                  if(Math.abs(x - base.x) < 0.001 && Math.abs(y - base.y) < 0.001){
                        desc+= "_correct";
                  }
                  long shadowID = dbb.insertPoint(new Point2D.Double(x,y), nodeID +
"_" + i + desc, Segment.SHADOW);
                  Node2D nd = new Node2D(x, y, shadowID, Segment.SHADOW);
                  nd.setParent(base);
                  res.add(nd);
                  i++;
                  k+=2;
            }
        }
        if(c != null && !c.isClosed()){
              c.close();
        }
        return res;
   }

   public double [][] fillOutMatrix (DBbridge dbb, Segment type) throws Exception{
        // if type anchor - anchors to nodes distances, otherwise node to node
        // Since we use Linked list, we rely on Node order and array index to
identify distances.
        List<Integer> ls1 = null;
        List<Integer> ls2 = null;
        if(type.getDbid() == Segment.NODE.getDbid()){
              ls1 = ls2 = dbb.getNodeIDtList(type);
        }
        else if(type.getDbid() == Segment.ANCHOR.getDbid()){
              ls1 = dbb.getNodeIDtList(type);
              ls2 = dbb.getNodeIDtList(Segment.NODE);
        }
        else{
              throw new Exception("this segment type is not supported yet");
        }
```

```
        int k = ls1.size();
        int l = ls2.size();
        double result [][] = new double[k][l];
        for(int i = 0; i < k; i++){
            for(int j = 0; j < l; j++){
                //find distance between id's i and j, fill [i][j]
                double dist = dbb.getDistance(ls1.get(i), ls2.get(j));
                result[i][j] = dist;
            }
        }
        return result;
    }

    public static Point2D PointOnCircle(double radius, double angleInDegrees,
Point2D origin)
    {
        // Convert from degrees to radians via multiplication by PI/180
        double x = (radius * Math.cos(angleInDegrees * Math.PI / 180F)) +
origin.getX();
        double y = (radius * Math.sin(angleInDegrees * Math.PI / 180F)) +
origin.getY();
        return new Point2D.Double(x, y);
    }
}
```

### 3.4 Geometrical static methods

```
package org.tsi.extended;
import java.awt.geom.Point2D;

public class MyGeom {

    public static Point2D getArcCenter(Point2D p1, Point2D p2, Point2D p3){
        double x1 = p1.getX();
        double y1 = p1.getY();
        double x2 = p2.getX();
        double y2 = p2.getY();
        double x3 = p3.getX();
        double y3 = p3.getY();
        double x21 = x2-x1; double y21 = y2-y1;
        double x31 = x3-x1; double y31 = y3-y1;
```

```
        double h21 = Math.pow(x21,2)+Math.pow(y21,2); double h31 =
Math.pow(x31,2)+Math.pow(y31,2);
        double d = 2*(x21*y31-x31*y21);
        double a = x1+(h21*y31-h31*y21)/d;
        double b = y1-(h21*x31-h31*x21)/d;
        Point2D res = new Point2D.Double(a, b);
        return res;
    }

    public static double getArcRadius(Point2D p1, Point2D p2, Point2D p3){
        double x1 = p1.getX();
        double y1 = p1.getY();
        double x2 = p2.getX();
        double y2 = p2.getY();
        double x3 = p3.getX();
        double y3 = p3.getY();
        double x21 = x2-x1; double y21 = y2-y1;
        double x31 = x3-x1; double y31 = y3-y1;
        double h21 = Math.pow(x21,2)+Math.pow(y21,2); double h31 =
Math.pow(x31,2)+Math.pow(y31,2);
        double d = 2*(x21*y31-x31*y21);
        double r = Math.sqrt(h21*h31*(Math.pow(x3-x2,2)+Math.pow(y3-
y2,2)))/Math.abs(d);
        return r;
    }

    public static double getAngleOfPointOnCircle(Point2D p, Point2D center, double
radius) {
        // returns the angle in degrees in the interval (0°, 360°) between the 12-
hour point and the second point
            double x = p.getX();
            double y = p.getY();
            // calculate the circle radius
            // double radius = Math.sqrt(Math.abs(x - centerX) * Math.abs(x -
centerX) + Math.abs(y - centerY) * Math.abs(y - centerY));
            // calculate the coordinates for the 12-hour point on the circle
            double p0x = center.getX();
            double p0y = center.getY() - radius;
            // calculate and return the angle in degrees in the range 0..360
            return (2 * Math.atan2(y - p0y, x - p0x)) * 180 / Math.PI;
        }
```

```
    public static Point2D getPointAt(Point2D center, double radius, double angle) {
         angle *= Math.PI / 180;
         double x = center.getX() + Math.sin(Math.PI - angle) * radius;
         double y = center.getY() + Math.cos(Math.PI - angle) * radius;
        return new Point2D.Double(x, y);
    }
}
```